

FEBRERO 1985 - 250 PTAS.
Todospectrum
AÑO 1 - NUMERO 6.
REVISTA EXCLUSIVA PARA USUARIOS
GHOSTBUSTERS
Lo mejor en representación de funciones
LOGO: Tortugas para todos
Programas de gestión a su medida. Analizamos el SITI
Interrupciones del Z-80
Juego de los espejos, vocales acentuadas, intelect...



Portada: DINAMIC

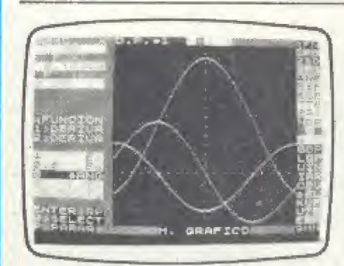
DIRECTOR:
Simeón Cruz González
COORDINADOR EDITORIAL:
Emiliano Juárez
REDACCION:
Juan Arencibia, Fernando García, José C. Tomás, Gumersindo García, Luis M. Brugarolas, Ricardo García, David García
DISEÑO: Ricardo Segura

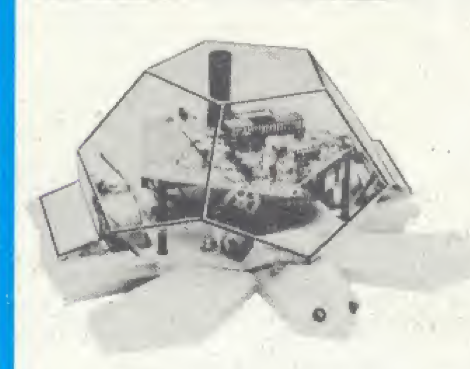
Editado por
PUBLINFORMATICA, S. A.
Presidente: Fernando Bolín
Director Editorial: Norberto Gallego

Administración:
INFODIS, S. A.
Gerente de Circulación y ventas:
Luis Carrero
Producción:
Miguel Onieva
Director de Marketing:
Antonio González
Servicio al cliente:
Julia González. Tel. 733 79 69
Administración:
Miguel Atance y Antonio Torres
Jefe de Publicidad
María José Martín
Dirección y redacción:
Bravo Murillo, 377-5.º A. Tel. 733 74 13
Telex: 48877 OPZX e 28020 Madrid
Administración y Publicidad:
Bravo Murillo, 377-3 E. Tels. 733 96 62/96
Publicidad Madrid:
María Lavalle
Publicidad Barcelona:
María del Carmen Ríos, Jorge González. Pelayo, 12. Tel. (93) 301 47 00. Ext. 27 y 28. 08001 Barcelona.
Depósito legal: M-29041-1984
Distribuye S.G.E.L.
Avda. Valdelaparra, s/n. Alcobendas-Madrid.
Fotomecánica: Karmat, C/ Pantoja, 10. Madrid.
Fotocomposición: Artecomp.
Imprime: Héroes, C/ Torrelara, 8. Madrid.
Esta publicación es miembro de la Asociación de Revistas de Información ARI, asociada a la Federación Internacional de Prensa Periódica, FIPP.

SUSCRIPCIONES:
Rogamos dirijan toda la correspondencia relacionada con suscripciones a:
TODOSPECTRUM
EDISA: Tel. 415 97 12
C/ López de Hoyos, 141-5.º
28002 MADRID
(Para todos los pagos reseñar solamente TODOSPECTRUM)
Para la compra de ejemplares atrasados diríjanse a la propia editorial
TODOSPECTRUM
C/ Bravo Murillo, 377-5.º A
Tel. 733 74 13 - 28020 MADRID

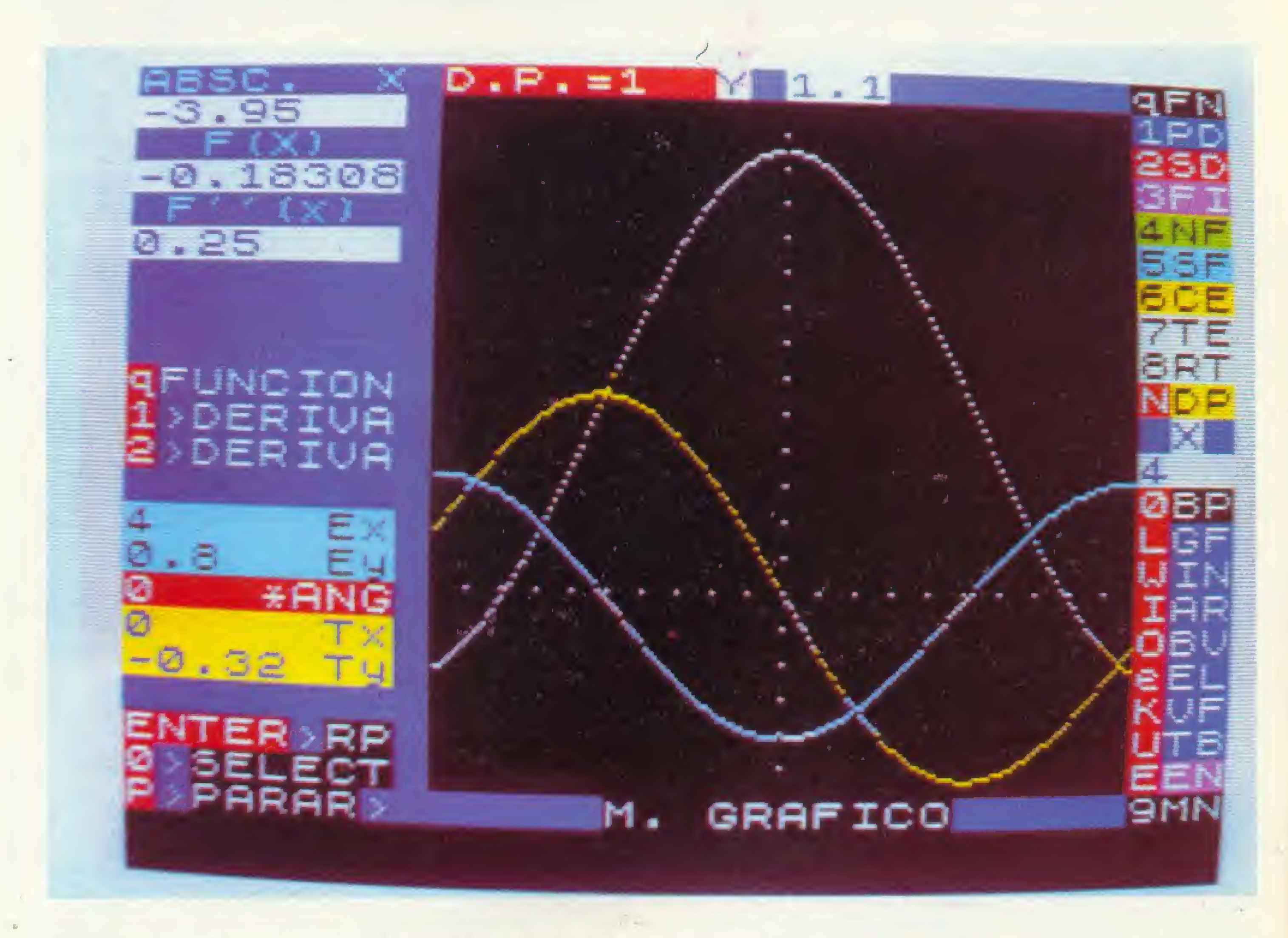
AÑO I • NUMERO 





Si se nos permite, este número seis de TODOSPECTRUM queremos dedicarlo a cuantos han confiado en la idea de esta revista. Cumplimos medio año y queremos celebrarlo con algunos cambios que no escaparán a la atención de los asiduos lectores. Un sumario más claro, una doble página dedicada a los juegos de mayor éxito, y "un corcho" para anotar vuestros anuncios, entre otros cambios.

Representación



Nombrar a Marbella es nombrar al turismo, como bien lo saben los malagueños, preocupados por el estudio del inglés que les facilite el puesto de trabajo. A la fiebre del turismo que amplía el número de habitantes de 80.000 a 300.000 en verano, se ha unido recientemente la imperable fiebre del Spectrum. Una pequeña academia especializada en matemáticas y física para estudiantes de BUP y COU, ha incorporado el Spectrum. TODOSPECTRUM le ofrece su mejor programa.

Este programa está pensado y realizado para ayudar a estudiar con menos esfuerzo una parte importante de la teoría de funciones, siendo sus aplicaciones didácticas muy amplias y su uso aconsejable a profesores y alumnos de BUP-COU.

Empecemos dando una definición sencilla. Una función (real de variable real) es una expresión matemática que transforma un número real en otro:

F(x)

X → Y con X, Y

pertenecientes al conjunto R.

Para representarla nos ayuda-

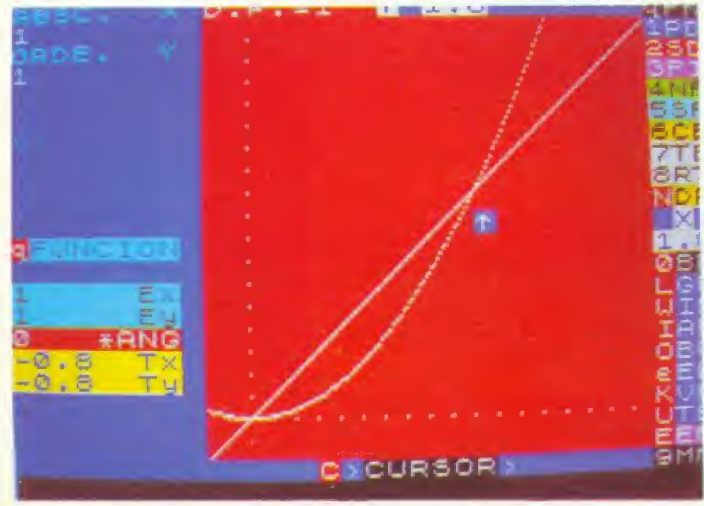
de funciones

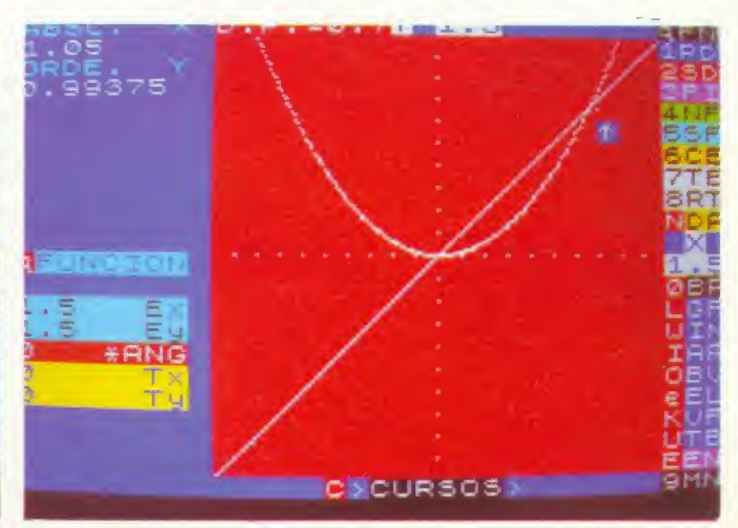
mos de dos ejes (líneas rectas) formados por infinitos puntos alineados, representando cada uno de ellos a un número R, y situándolas perpendicularmente, cortándose en un punto llamado origen de coordenadas (0,0). Buscaremos el punto correspondiente al número real x en el eje OX y el correspondiente al número Y=F(x) en el eje OY.

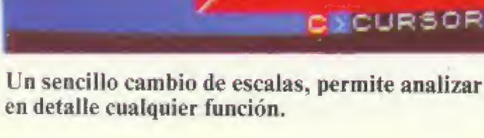
Este par de números ordenados definen un punto del plano R de coordenadas (X, Y). Al conjunto de puntos obtenido dando a X valores reales se llama GRAFO de la función F(x). Para una función dada, puede haber números reales X (puntos) que no tengan transformados (imágenes de X), es decir: X → ?; en este caso en el grafo habrá puntos o conjuntos de puntos que faltarán, los ordenadores y calculadoras se bloquean (dando error) ya que la representación es imposible en estos intervalos (que no pertenecen al dominio de F(x)). Este programa resuelve el problema de distinta manera según el tipo de función que estemos estudiando. Para funciones como LNx, cuyo dominio de definición lo forman los números reales positivos, el programa dispone los ejes de manera que la representación gráfica se realice sólo para estos valores de X mediante traslación de ejes. En el caso de funciones tales como 1/x SEN x/x, cuyo dominio está formado por todos los números reales excepto el cero, el programa saltará todos los valores que hagan al denominador menor que una millonésima. Es decir, para funciones que tengan asíntotas verticales (F(x) tiende a infinito), el ordenador evitará darle esos valores, con lo cual queda resuelto el problema. Expliquemos el funcionamiento con ejemplos prácticos.

Vamos a empezar nuestra andadura desde el principio, suponiendo que ya hemos «tecleado» el programa y lo tenemos copiado en una cinta. Búsquelo e introduzca la sentencia LOAD "". Le preguntará una vez terminada su carga, por el color del papel en el que se va a realizar el grafo de la función y aparecerá en la pantalla el menú resumen. En él podrá observar dos partes: la superior donde están indicadas las teclas utilizadas y su disposición en el ordenador y la inferior donde se indica el uso de las mismas. En las dos últimas líneas de la pantalla verá: «S SELECTOR». Llamamos selector a la parte del programa que puede manipular los parámetros o valores que hacen falta para la representación. Llamamos modo GRAFICO a la parte donde se realiza ésta. para pasar a éste último, desde el selector, basta pulsar ENTER. Pero antes, después de pulsar ((S)), tendremos que darle al ordenador los datos que nos pide. Nos preguntará por el numerador; introduzcamos, por ejemplo, SEN x. Después el denominador: x. No ponga ((/)), sólo ENTER. Si cometió algún error al introducir el numerador, cuando le pida el denominador introduzca "0", esto lo entenderá como que desea cambiar el numerador. Si la función es un polinomio o no posee denominador, hágale valer uno pulsando directamente ENTER.

Si posteriormente desea cambiar la función, pulse ((4)) cuando el modo sea selector. Si ésta posee potencias de x se deben introducir en forma de producto reiterado para valores negativos de X y en su





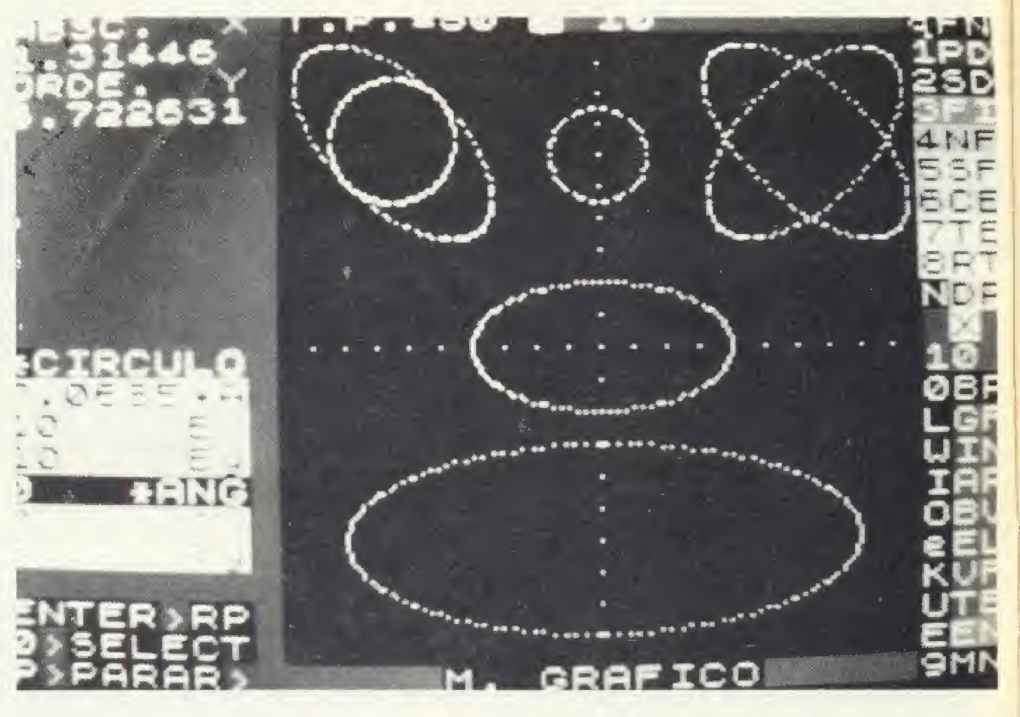

Un sencillo cambio de escalas, permite analizar en detalle cualquier función.

forma «BASIC» (↑) para valores positivos. Es decir: x elevado al cubo entre −10 y 10 debe expresarse como x*x*x y si se representa, utilizando la traslación, entre 0 y 20 se puede introducir x↑3. En el numerador no se pueden utilizar denominadores que tiendan a cero en el intervalo de representación: NUMER F(x) = TAN x; DENOM F(x) = 1, bloqueará el programa. Sin embargo, haciéndolo de esta otra forma no:

NUMER F(x) = SEN x.

DENOM F(x) = COS x.

Después de introducida la función sen x/x, nos pedirá los valores entre los que queremos representar. Nos pedirá dos: uno para los valores de x (ESCALA X Ex) y otro para F(x) (ESCALA Y-Ey). Introduzcamos: Ex=30; Ey=1. En este caso la variable independiente x tomará valores comprendidos entre −30 a 30 y el punto imagen entrará en pantalla si está comprendido entre −1 y 1. Ahora si pulsa ENTER pasará al modo gráfico y se realizará la representación gráfica. En la parte superior de la pantalla observará un número en «papel» rojo donde pone: D.P.=2. Esto es la Densidad de Puntos y marca el incremento que queremos que tenga la X. ahora, con valor dos, el incremento de x=30*2/80; en general, si llamamos N a la densidad de puntos, se obtendrá un incremento de x=Ex*N/80. La densidad de puntos se cambia pulsando ((N)) en modo selector. Si desea cambiar en el mismo modo. Se pueden introducir operaciones indicadas como "2*PI" y si comete error al introducir Ex, meta "0" en Ey, pidiéndole de nuevo Ex. Resumiendo (en modo selector):



MENU ((M)) – ((9))

CAMBIO DE F(X) –((4))

CAMBIO DE ESCALAS –((6))

CAMBIO DE DENSIDAD P. –((N))

Nos podríamos hacer ahora una buena pregunta: ¿Estoy obligado a ver siempre el grafo de la función

```
 100 REM **********************
 110 REM ANDRES JESUS SANCHEZ
 120 REM MANOLI FERNANDEZ G.
 130 REM **********************
 180 BRIGHT 0
 190 CLEAR
 200 INPUT "COLOR PAPEL ? (0-3)
";pa
 210 IF pa>3 OR pa<0 THEN  BEEP
.5,0: GO TO 200
 220 POKE 23609,10
 230 REM ********DEF FN******
 240 DIM l$(11,8)
 250 DIM b$(5,8)
 260 DIM e$(2,3)
 270 DIM a$(1,3)
 280 DIM x(5)
 290 DIM p$(2,8)
 300 DIM g$(1,8)
 310 DIM o$(1,8)
 320 DEF FN n(x)=VAL n$
 330 DEF FN d(x)=VAL d$
 340 DEF FN f(x)=FN n(x)/FN d(x)
+j
 350 DEF FN g(x)=(FN f(x+h)-FN f
(x))/h+j
 360 DEF FN h(x)=(FN g(x+h)-FN g
(x))/h+j
 370 GO SUB 2710
 380 GO SUB 2630
 390 GO TO 830
 400 REM ****GRAFO ***
 410 FOR n=-80 TO 80 STEP b
 420 LET x=n*s/80
 430 IF ABS FN d(x-i)<1E-6 THEN
 GO TO 490
 440 IF dd=0 THEN  LET y=FN f(x-
i)
 450 IF dd=1 THEN  LET y=FN g(x-
i)
 460 IF dd=2 THEN  LET y=FN h(x-
i)
 470 IF ABS y<k AND ABS x<s THEN
  PLOT  INK 7-dd;152+n,87+y*80/k
 480 IF PEEK z=48 THEN  POKE z,6
4: PRINT AT 20,0;"          ": GO
TO 820
 490 NEXT n
 500 GO TO 780
 510 REM *** M. GRAFICO***
 520 FOR n=-80 TO 80 STEP b
 530 IF PEEK z=108 AND rr=0 AND
dd<>3 AND f<>1 AND gt<>1 THEN  L
ET rp=1: POKE z,64: FOR v=0 TO 9
: PRINT AT v,0;"          ": NEXT
v: PRINT AT 2,0; BRIGHT 1;"L>GRA
FO>": IF x(1)<>0 OR x(2)<>0 OR x
(3)<>0 OR x(4)<>0 OR x(5)<>0 THE
N  FOR v=1 TO 5: LET b$(v,1 TO 3
)="X"+STR$ v+"=": LET b$(v,4 TO
8)=STR$ x(v): PRINT AT 3+v,0; PA
PER 3; INK 1;b$(v): NEXT v
 540 IF rp=1 THEN  PRINT AT 19,0
;"          ";AT 21,0;"          ":
GO TO 420
 550 LET x=(n*s/80)
 560 IF ABS FN d(x-i)<1E-6 THEN
 GO TO 760
 570 IF dd=0 OR dd=3 THEN  LET y
=FN f(x-i)
 580 IF dd=1 THEN  LET y=FN g(x-
i)
 590 IF dd=2 THEN  LET y=FN h(x-
i)
 600 IF dd=3 THEN  LET u=x: LET
x=FN f(x)+i: LET y=u-j: IF rr=0
THEN  LET p$(1)=STR$ (INT ((x-i)
*1E3+.5)/1E3): LET p$(2)=STR$ (I
NT (y*1E3+.5)/1E3): PRINT  INVER
SE 1;AT 1,0;p$(1);AT 3,0;p$(2)
 610 IF rr<>0 THEN  LET yy=y: LE
T xx=x: LET y=x*SIN rr+y*COS rr:
 LET x=x*COS rr-yy*SIN rr: LET p
$(1)=STR$ (INT (x*1E6+.5)/1E6):
LET p$(2)=STR$ (INT (y*1E6+.5)/1
E6): PRINT AT 1,0;p$(1);AT 3,0;p
$(2)
 620 IF f=1 AND x>=x2 AND x<x3 T
```

en los alrededores del origen de coordenadas? La respuesta es no. Disponemos de la tecla ((7)) que nos permite mover la función sobre la pantalla de TV. A esto lo hemos llamado TRASLACION. Si en la función que tenemos en estudio, (sen x/x), con escalas Ex = 30 y Ey = 1, introducimos un cambio en ((6)), haciendo Ex = 10 y Ey = .7 y trasladamos la función mediante ((7)), haciendo Tx = –9 y Ty = –.35, veríamos que los ejes y la función se han desplazado (trasladado) hacia el punto (–9, –.35) con referencia a la escala. Pruebe para otros valores:

Tx = 9 y Ty = –.35 etc.

Si cuando le pregunta Tx pulsa ENTER y hace lo mismo con Ty, entonces Tx y Ty tomarán el valor cero.

Con este programa se puede provocar también la rotación de F(x) alrededor del centro de la pantalla. Esto se consigue pulsando ((8)) e introduciendo un ángulo en grados. Si introduce ENTER como valor del ángulo, éste valdrá cero. Pruebe: TECLA

((4)) NUM = x*x ; DEN = 5
((6)) Ex = 5 ; Ey = 5
((7)) Tx = 1 ; Ty = –1
((8)) ANGULO = 90

Déle otros valores al ángulo como 180, –90, etc. Pulse ENTER cuando quiera representar.

Todo lo mencionado se puede ejecutar en el modo selector.

¿Qué se puede hacer mientras se lleva a cabo la representación? Tres cosas:

1. Pulsando ((P)) se paraliza la ejecución de ésta, quedando un valor de x y su correspondiente F(x) impreso en la pantalla. Si pulsamos ((P)) de nuevo le daremos a x el valor siguiente y así sucesivamente. Para continuar la representación pulsar, por ejemplo ((J)).
2. Pulsando ((ENTER)) aumentaremos el incremento de x, representando menos puntos y más rápidamente. Se desactiva pulsando, por ejemplo ((J)).
3. Pulsando ((0)) dejamos de representar y volvemos al modo selector donde, si pulsamos de nuevo ((0)), borrará. MODO ((0)).

Gráfico (nos lleva al selector-borra pantalla)

Mientras dura la representación van apareciendo los valores de x y de F(x), así como los valores que poseen, en ese momento, Ex, Ey,



Tx, Ty, ANG, Función, N, etc. Si no tenemos mucho interés en ver los valores de x y de F(x) podemos ejecutarla más rápidamente pulsando ((L)). A esta orden la hemos llamado GRAFO, ya que es eso lo que hace sin más. Esta orden se puede ejecutar tanto desde el modo gráfico como desde el selector. Es irreversible, es decir, una vez pulsada ((L)) en modo gráfico no se puede desactivar. En modo selector se puede hacer pulsándola de nuevo. Dándole a N, en ((N)), valores grandes (de 4 a 10 por ejemplo) y pulsando ((L)) se pueden conseguir representaciones de poca densidad de puntos pero rápidamente ejecutadas. Esta orden

```
HEN  LET ff=ff+ABS y*s/80: LET 1
$(4,1 TO 6)=STR$ (INT (ff*10+.5)
/10): LET 1$(4,7 TO 8)="*A": PRI
NT AT 7,0; PAPER 3;1$(4): IF ABS
 y<=k THEN  POKE 23677,152+x*80/
s: POKE 23678,87: DRAW  INK 7-dd
;0,y*80/k
 630 IF f=1 AND x>=x2 AND x<x3 A
ND ABS y>k THEN  POKE 23677,152+
x*80/s: POKE 23678,87: DRAW  INK
 7-dd;0,80*SGN y
 640 IF f=1 THEN  LET p$(1)=STR$
 (INT ((x)*1E6+.5)/1E6): LET p$(
2)=STR$ (INT ((FN f(x-i))*1E6+.5
)/1E6): PRINT AT 1,0;p$(1);AT 3,
0;p$(2)
 650 IF ABS y<k AND ABS x<s THEN
  PLOT  INK 7-dd;152+x*80/s,87+y
*80/k
 660 IF dd<>3 AND rf=0 THEN  LET
 p$(2)=STR$ (INT (((FN f(x-i))-j
)*1E6+.5)/1E6): LET p$(1)=STR$ (
INT ((x-i)*1E6+.5)/1E6): PRINT
INVERSE 1;AT 1,0;p$(1);AT 3,0;p$
(2)
 670 IF dd=1 AND rf=0 THEN  LET
g$(1)=STR$ (INT (((FN g(x-i))-j)
*1E2+.5)/1E2): PRINT  INVERSE 1;
AT 5,0;g$(1)
 680 IF dd=1 AND f=1 THEN  LET g
$(1)=STR$ (INT (y*1E2+.5)/1E2):
PRINT AT 5,0; INVERSE 1;g$(1)
 690 IF dd=2 AND rf=0 THEN  LET
o$(1)=STR$ (INT (((FN h(x-i))-j)
*1E2+.5)/1E2): PRINT  INVERSE 1;
AT 5,0;o$(1)
 700 IF dd=2 AND f=1 THEN  LET o
$(1)=STR$ (INT (y*1E2+.5)/1E2):
PRINT  INVERSE 1;AT 5,0;o$(1)
 710 IF gt=1 AND n>=(x1+i)*80/s
AND x1+i<>s+i THEN  POKE z,52: L
ET cg=1: PRINT AT 19,0;"
";AT 20,0;"               ";AT 21,0;"
      ": GO TO 1240
 720 IF PEEK z=13 THEN  LET n=n+
3: PRINT AT 0,18;"E"
 730 IF PEEK z<>13 THEN  PRINT A
T 0,18;" "
 740 IF PEEK z=112 THEN  PAUSE 0
 750 IF PEEK z=48 THEN  PRINT AT
 1,0;"           ";AT 3,0;"
";AT 5,0;"            ";AT 19,0;"
      ";AT 20,0;"           ";AT 21,
0;"          ";AT 1,0: LET p=1: LE
T gt=0: LET ggtt=0: LET ff=0: GO
 TO 830
 760 IF p=1 THEN  INPUT "": PRIN
T #0;"                ";  INVERSE 1;"
PULSA J ":  PAUSE 0: PRINT AT 1,
0;"           ";AT 3,0;"           ";A
T 5,0;"            ": GO TO 1160
 770 NEXT n
 780 REM **REPONE VARIABLES**
 790 IF rp<>1 THEN  PRINT AT 1,0
;"           ";AT 3,0;"            ";AT
 5,0;"            ";AT 10,0;"
 "
 800 PRINT AT 19,0;"            ";AT
 20,0;"           ";AT 21,0;"
  ";AT 0,18;" "
 810 IF f=1 THEN  BEEP .1,-20: B
EEP .1,-20: INPUT "": PRINT #0;"
              "; FLASH 1; INVERSE 1;
" PULSE  J ": PAUSE 0
 820 LET ss=0: LET f=0: LET ff=0
: LET gg=0: LET sf=0: LET ggtt=0
: LET gt=0: LET p=1: LET qq=0: L
ET e=1
 830 POKE z,64: BEEP .05,0: BEEP
 .05,0: FOR n=7 TO 1 STEP -1: PA
USE 3: BORDER n: NEXT n: GO TO 1
650
 840 REM *** TABLA VALORES ***
 850 CLS
 860 LET dd=0: LET p=0: LET y$="
0": LET sa=0
 870 PRINT AT 6,8; INVERSE 1;"TA
```

es incompatible con la rotación ((8)). Es muy práctico cuando lo que deseamos es tener una idea sobre la forma del grafo de la función sin que nos preocupe los valores que toma.

Con lo explicado hasta el momento ya sería suficiente para que este programa tuviese aplicaciones importantes: Teorema de Taylor, de Rolle, del valor medio, etc. ¿Qué podemos hacer si lo que queremos es dibujar o estudiar los valores que toman la primera derivada (F' (x)) o la segunda derivada (F'' (x)) o la función inversa (Finv (x)) de la función F (x)? Podrían calcularse éstas e introducirse en ((4)) pero esto no es necesario:

Si pulsa ((1)) tendrá a su disposición la primera derivada; si pulsa ((2)) la segunda; si pulsa ((3)) la función inversa y en ((q)) la función F (x).

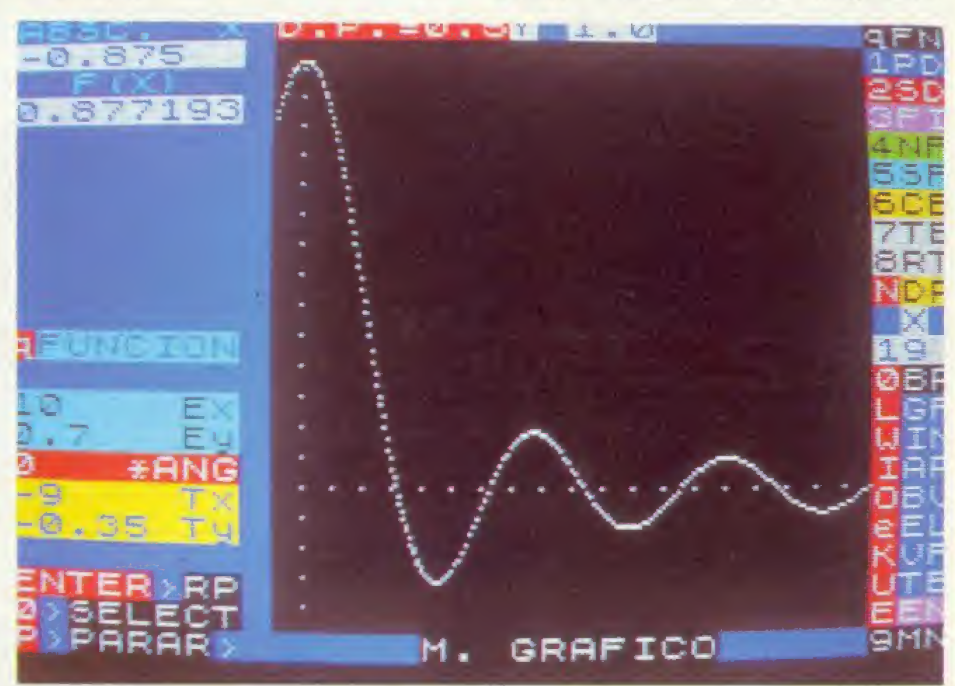
Para la función inversa no actúa ((L)). Las órdenes que no compatibiliza el programa las puede ir viendo sobre la práctica.

Si elige ((q)) dibujar la función con tinta blanca. Si elige ((1)), la derivada, con tinta amarilla. Si elige ((2)), la segunda derivada, con tinta cyan y si elige ((3)), la función inversa, con tinta verde. Esto nos permitirá estudiar todo lo relativo a máximos y mínimos, crecimiento, concavidad, puntos de inflexión, etc., de la función F (x) introducida en ((4)). Es aconsejable, para rotaciones y para la Finv (x), tomar Ex = Ey.

Ejemplo:

En ((4)) NUM = SEN ; DEN = x
En ((6)) Ex = 4 ; Ey = .8
En ((7)) Tx = 0 ; Ty = -.32
En ((N)) D.P. = 1
En ((8)) ANG = ENTER < > Ang = 0
En ((q)) Para elegir F (x)

Pulse ahora ENTER y la representación de F(x) se llevará a término. Cuando haya finalizado




```
BLA DE VALORES": PRINT AT 11,8;"
ABSC. X";AT 13,12;"ORDE. Y";AT 1
5,16;"DERIVADA"
 880 PLOT 20,20: DRAW 215,0: DRA
W 0,135: DRAW -215,0: DRAW 0,-13
5
 890 INPUT  FLASH 1;" INCREMENTO
 X? ";" ";l1: LET l=80*l1/s: CLS

 900 IF l=0 THEN  GO TO 1050
 910 PRINT #0; PAPER 2;" ABSC.
X    ORDE. Y     1 DERIV."
 920 FOR n=-80 TO 80 STEP 1
 930 LET sa=sa+1
 940 LET x=n*s/80
 950 IF ABS FN d(x-i)<1E-6 THEN
 GO TO 1020
 960 LET p$(1)=STR$ (INT ((x-i)*
1E6+.5)/1E6): LET p$(2)=STR$ (IN
T ((FN f(x-i)-j)*1E6+.5)/1E6): L
ET g$(1)=STR$ (INT ((FN g(x-i)-j
)*1E3+.5)/1E3): PRINT TAB 2;p$(1
);TAB 12;p$(2);TAB 22;g$(1)
 970 BEEP .03,12*SGN (FN g(x-i)-
j)
 980 IF sa=22 THEN  LET p=1: INP
UT "": PRINT #0; INVERSE 1;"  Co
ntinuo ?-(s/n)  ": PAUSE 0
 990 IF p=1 AND PEEK z=115 THEN
 CLS : LET sa=0: INPUT "": PRINT
 #0; PAPER 2;
  " ABSC. X    ORDE. Y    1 DERI
V.  "
1000 IF p=1 AND PEEK z<>115 THEN
  CLS : POKE z,64: GO SUB 2630:
GO TO 1650
1010 LET p=0
1020 NEXT n
1030 INPUT "": PRINT #0; INVERSE
 1; FLASH 1;" DE NUEVO ? (s/n) "
: PAUSE 0
1040 IF PEEK z=115 THEN  CLS : G
O TO 850
1050 IF PEEK z<>115 THEN  POKE z
,64: CLS : LET p=1: GO SUB 2630:
 GO TO 1650
1060 REM **** PARAMETROS***
1070 FOR x=1 TO 5
1080 PRINT AT 21,0; FLASH 1;"  X
(";x;")  ": INPUT y$
1090 IF y$="" THEN  GO TO 1120
1100 LET x(x)=VAL y$
1110 LET b$(x)=STR$ x(x)
1120 NEXT x
1130 PRINT AT 21,0;"           "
1140 POKE z,64: LET p=1: LET qq=
0
1150 RETURN
1160 REM ******* SELECTOR ****
1170 POKE 23658,0
1180 IF PEEK z=120 THEN  GO SUB
1060
1190 IF PEEK z=110 THEN  PRINT
PAPER 6; INK 0;AT 21,0; FLASH 1;
"D. PUNTO": INPUT b: PRINT AT 21
,0;"            ": LET qq=0: LET p=1
: POKE z,64: IF b=0 THEN .POKE z
,110: GO TO 5015
1200 IF PEEK z=107 THEN  PRINT A
T 21,0; INVERSE 1; FLASH 1; BRIG
HT 1;"ABSC. X?": PRINT  INK 5;AT
 0,0;"ABSC.  X";AT 2,0;"ORDE.  Y
": INPUT x: LET x=x+i: PRINT AT
21,0;"            ": LET rr=0: LET r
$="0": LET p=1: POKE z,64: GO TO
 560
1210 IF PEEK z=108 AND rp=0 THEN
  LET rp=1: LET f=0: LET rr=0: L
ET r$="0": LET gt=0: LET p=1: PO
KE z,64: IF dd=3 THEN  LET dd=0
1220 IF PEEK z=108 AND rp=1 THEN
  LET rp=0: LET p=1: POKE z,64
1230 IF PEEK z=111 THEN  LET r$=
"0": LET ss=0: LET ff=0: LET f=0
: LET p=1: LET gt=0: LET ggtt=0:
 LET dd=0: LET rr=0: LET i=0: LE
T j=0: LET qq=0: INPUT "": PRINT
 #0;" BORRO:7 8 i w  ###  DEJO:q
 6 ": PAUSE 100: BEEP .05,10: IN
PUT "": POKE z,64
1240 IF PEEK z=52 OR e=0 THEN  P
RINT  FLASH 1;AT 21,0; BRIGHT 1;
 INK 0;"NUM F(x)": INPUT ;n$: PR
```

ésta, pulse ((1)) seguido de ENTER y aparecerá el grafo de F'(x). Repita la operación para ((2)). Si desea efectuarla rápidamente pulse previamente ((L)). Finalizada la última gráfica tendremos en pantalla las tres representaciones simultáneamente.

Un estudio aparte requiere la representación de CONICAS ya que posee una pequeña diferencia con respecto al que se hace de las funciones en forma explícita. En éstas, puede cambiar los valores Ex, Ey, Tx, etc., cuantas veces quiera antes de representar la gráfica ya que no se realiza hasta que no damos la orden ENTER. En cónicas le pedirá el total de puntos a imprimir y no la D.P., además, le pedirá el semieje, mayor primero, el menor después y se ejecutará la gráfica sin esperar la orden ENTER. Además los ejes permanecerán en el centro de la pantalla aunque exista traslación, comportándose ésta de una manera diferente y lo que se desplaza ahora es la cónica y no los ejes. En pantalla tenemos también el valor de la excentricidad de la elipse o del área del círculo, según el caso. Preparando previamente los datos podemos dibujar la elipse trasladada y rotada en la pantalla.



Un apartado importante referente a las funciones es el relativo a

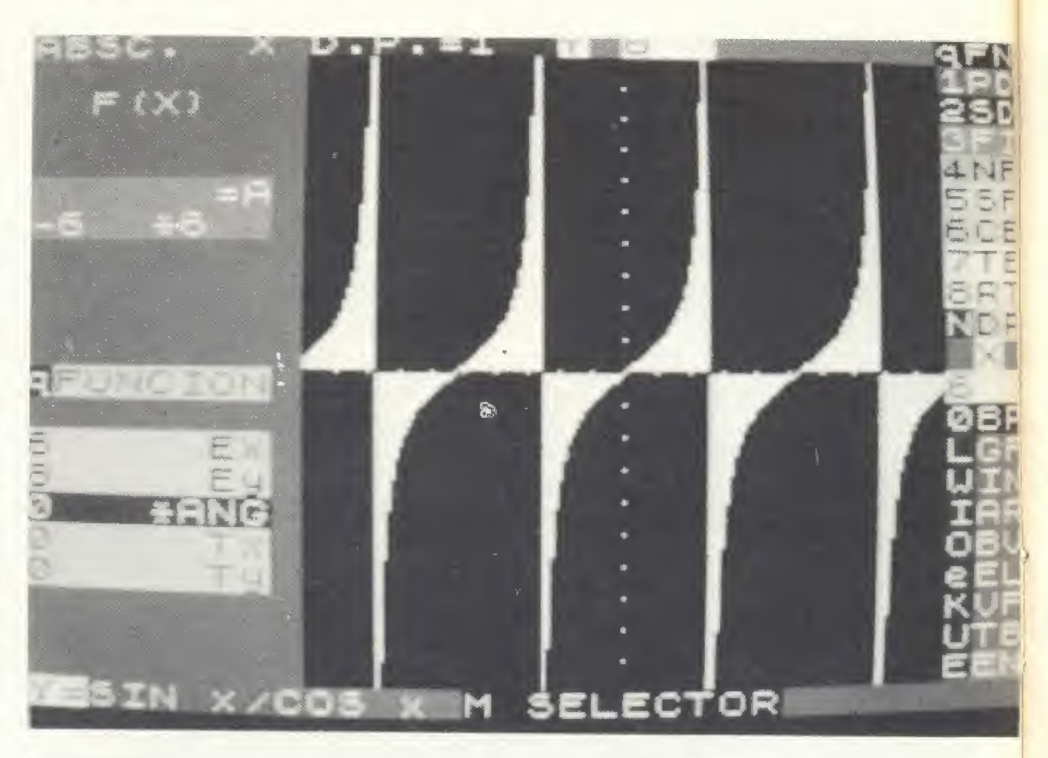
```
INT  INK 0; FLASH 1; BRIGHT 1;AT
 21,0;"DEN F(x)": INPUT ;d$: PRI
NT AT 21,0;"          ": LET qq=0:
 LET p=1: POKE z,64: IF gt=0 THE
N  GO TO 1300
1250 IF d$="0" THEN  POKE z,52:
GO TO 1240
1260 IF d$="" THEN  LET d$="1"
1270 IF gt=1 AND cg=1 THEN  PRIN
T AT 21,0; INVERSE 1; FLASH 1; B
RIGHT 1;"HASTA X?": INPUT x$: LE
T x1=VAL x$: LET rp=0: PRINT AT
21,0;"          ": POKE z,64: LET
cg=0
1280 IF gt=1 AND cg=1 THEN  LET
ii=i: LET jj=j: POKE z,64
1290 IF gt=1 AND cg=1 AND ggtt<>
0 THEN  GO TO 1160
1300 IF d$="0" THEN  POKE z,52:
GO TO 1240
1310 IF d$="" THEN  LET d$="1"
1320 IF e=0 THEN  LET s1=10: LET
 k1=10
1330 IF PEEK z=54 OR e=0 THEN  P
RINT  FLASH 1; BRIGHT 1;AT 21,0;
 INK 0;"ESCALA X": INPUT ;s$: PR
INT  FLASH 1; INK 0; BRIGHT 1;AT
 21,0;"ESCALA Y": INPUT ;k$: PRI
NT AT 21,0;"          ": LET s=ABS
 (VAL s$): LET k=ABS (VAL k$): L
ET e=1: LET p=1: LET qq=0: POKE
z,64
1340 IF s=0 OR k=0 THEN  POKE z,
54: GO TO 1330
1350 IF PEEK z=113 THEN  LET dd=
0: LET qq=0: LET p=1: POKE z,64
1360 IF PEEK z=49 THEN  LET dd=1
: LET qq=0: LET p=1: POKE z,64
1370 IF PEEK z=50 THEN  LET dd=2
: LET qq=0: LET p=1: POKE z,64
1380 IF PEEK z<>56 THEN  GO TO 1
420
1390 PRINT  FLASH 1; BRIGHT 1;AT
 21,0; INVERSE 1;"ANGULO  ": INP
UT ;r$: PRINT AT 21,0;"          "
: LET qq=0: LET rp=0: LET p=1: P
OKE z,64
1400 IF r$="" THEN  LET r$="0"
1410 LET rr=VAL r$*PI/180: IF rr
<>0 THEN  LET f=0: LET ff=0
1420 IF PEEK z=119 THEN  LET gt=
1: LET rp=0: LET cg=1: LET p=1:
POKE z,52: GO TO 1240
1430 IF PEEK z<>55 THEN  GO TO 1
480
1440 PRINT  FLASH 1;AT 21,0; BRI
GHT 1; INK 0;"  TRASL. X": LET p
=1: INPUT ;i$: PRINT  BRIGHT 1;
FLASH 1;AT 21,0; INK 0;"
TRASL. Y": INPUT ;j$: PRINT AT 2
1,0;"          ": LET qq=0: POKE z
,64: IF dd=3 THEN  LET dd=0
1450 IF i$="" THEN  LET i$="0"
1460 IF j$="" THEN  LET j$="0"
1470 LET i=VAL i$: LET j=VAL j$
1480 IF PEEK z=51 THEN  LET dd=3
: LET rp=0: LET i=0: LET j=0: LE
T f=0: LET ff=0: LET p=1: LET qq
=0: POKE z,64
1490 IF PEEK z<>105 THEN  GO TO
1530
1500 LET f=1: LET rp=0: LET rr=0
: LET r$="0": PRINT AT 21,0; INV
ERSE 1; FLASH 1; BRIGHT 1;"DESDE
 X?": INPUT h$: PRINT AT 21,0; I
NVERSE 1; FLASH 1; BRIGHT 1;"HAS
TA X?": INPUT w$: LET p=1: PRINT
 AT 21,0;"          ": POKE z,64:
IF dd=3 THEN  LET dd=0
1510 IF h$="" AND w$="" THEN  LE
T x2=-s: LET x3=s: GO TO 1530
1520 LET x2=VAL h$: LET x3=VAL w
$
1600 IF PEEK z=101 THEN  PRINT A
T 11,0;"e>CONICA";AT 0,18;" ": L
ET rp=0: PRINT AT 21,0; PAPER 6;
 INK 0; FLASH 1;"To PUNTO": INPU
T b: PRINT AT 21,0;"          ": G
O TO 2340
1610 IF PEEK z=99 THEN  PRINT AT
 21,14; FLASH 1; BRIGHT 0;" C>CU
RSOR>": GO TO 2240
1620 IF PEEK z=117 THEN  LET c=0
```

la INTEGRAL. La integral definida entre dos límites representa el área comprendida entre la función y el eje de abscisas (eje OX). Para hacerla, pulse ((I)) y le pedirá los límites de integración. En caso de que quiera integrar entre los extremos de la pantalla, Ix = Ex y Ixx = Ex, pulse dos veces ENTER y calculará el área encerrada entre el grafo y el eje OX que se ve en pantalla. Si quiere cometer pocos errores debe hacer N lo más pequeño posible (N = .5 .1) aunque en este caso tardará más en el cálculo y la representación.

El comportamiento de los ejes sobre la traslación de los mismos en la integral, rotación, y función inversa es idéntico al que se explica para cónicas. En estos casos, cuando hay traslaciones, lo que se representa es F(X-Tx) + Ty referido a la posición que ocupa los ejes que aparecen en pantalla.

Ejemplo:

Introduzca en ((4)) la función – sen x/x, en ((6)) las escalas (Ex = 8; Ey = 1.2), en ((7)) la traslación (Tx = 0; Ty = 0), en ((8)) ANG = 0, en ((N)) la D.P. = N = 1, pulse ((I)) para reclamar la integral; intro-



NUMER	DENOM	Ex	Ey	Tx	Ty	ANG	D.P
x	x*x-3	5	5	0	0	0	1
x*x	5	5	5	0	0	90	1
COS x	.4	PI	PI	0	0	0	2
COS x	.4	PI	PI	0	0	180	2
x*x*x	1	1	1	0	0	0	2
x*x	x*x-4	5	5	0	0	0	1
x*x	x*x-4	5	5	-3	0	0	1
ATN x	x-2	2PI	1	0	0	0	1
1	x	3	3	-1	-1	0	.5

```
: LET cc=0: GO TO 840
1630 IF PEEK z=48 THEN  LET c=0:
 LET cc=0: LET p=1: LET qq=0: PO
KE z,64: GO SUB 2630
1640 REM **** CONTROL PANTALLA*
1650 IF (p=1 AND rp<>1) OR (p=1
AND rp=1 AND x(1)=0 AND x(2)=0 A
ND x(3)=0 AND x(4)=0 AND x(5)=0)
 THEN  FOR t=1 TO 5: PRINT AT 3+
t,0;"          ": NEXT t
1660 IF rr=0 AND f=0 THEN  LET r
f=0
1670 IF rr<>0 OR f=1 THEN  LET r
f=1
1680 IF p<>1 THEN  GO TO 1940
1685 IF p=1 THEN  PRINT AT 21,14
;"M SELECTOR"
1690 PRINT AT 0,9;"          ": PR
INT AT 0,9; BRIGHT 0;"D.P.=";: L
ET a$(1)=STR$ b: PRINT  PAPER 2;
a$(1)
1700 IF rp=0 THEN  PRINT AT 0,0;
 INK 5;"ABSC.  X"
1710 IF dd=0 THEN  PRINT AT 11,0
;"q"; INK 1;"FUNCION"
1720 IF rp=0 AND (rr=0 OR (rr<>0
 AND dd=0)) AND rp=0 AND dd<>3 T
HEN  PRINT AT 2,0; INK 5;"  F(X)
  "
1730 IF dd=1 THEN  PRINT AT 11,0
;"1>DERIVA"
1740 IF dd=1 AND rp=0 AND rr=0 T
HEN  PRINT AT 4,0; INK 5;"  F'(x
) "
1750 IF dd=1 AND rr<>0 THEN  PRI
NT AT 2,0; INK 5;"  F'(X) ";AT 4
,0;"        "
1760 IF dd=2 THEN  PRINT AT 11,0
;"2>DERIVA"
1770 IF dd=2 AND rp=0 AND rr=0 T
HEN  PRINT AT 4,0; INK 5;" F''(x
) "
1780 IF dd=2 AND rr<>0 THEN  PRI
NT AT 2,0; INK 5;" F''(x) ": PRI
NT AT 4,0;"        "
1790 IF rp=0 AND dd<>1 AND dd<>2
 THEN  PRINT AT 4,0;"          "
1800 IF dd=3 THEN  PRINT AT 11,0
;"3>F. INV": IF rp=0 THEN  PRINT
 AT 2,0; INK 3; BRIGHT 1;"F(inv)
 X"
1810 IF f<>1 AND rp<>1 THEN  PRI
NT AT 7,0;"          ";AT 8,0;"
    "
1820 IF gt=0 THEN  PRINT AT 10,0
;"        "
1830 IF rp=1 THEN  PRINT AT 0,0;
"          ";AT 2,0; BRIGHT 1;"L>G
RAFO>": IF x(1)<>0 OR x(2)<>0 OR
 x(3)<>0 OR x(4)<>0 OR x(5)<>0 T
HEN  FOR x=1 TO 5: LET b$(x,1 TO
 3)="X"+STR$ VAL "x"+"=": LET b$
(x,4 TO 8)=STR$ x(x): PRINT AT 3
+x,0; PAPER 3; INK 1;b$(x): NEXT
 x
1840 IF gt=1 THEN  PRINT AT 10,0
;"W>INTERV"
1850 IF f=1 THEN  LET l$(4,7 TO
8)="*A": LET l$(4,1 TO 6)=STR$ f
f: PRINT AT 7,0; PAPER 3;l$(4)
1860 IF f=1 THEN  LET l$(5,5)="*
": LET l$(5,1 TO 4)=STR$ x2: LET
 l$(5,6 TO )=STR$ x3: PRINT AT 8
,0; PAPER 3;l$(5)
1870 IF rf=0 THEN  LET e$(1)=STR
$ (s-i): LET e$(2)=STR$ (k-j): P
RINT  INVERSE 1;AT 0,19;e$(2);AT
 11,29;e$(1)
1880 IF rf=1 THEN  LET e$(1)=STR
$ s: LET e$(2)=STR$ k: PRINT AT
0,19;e$(2);AT 11,29;e$(1)
1890 LET l$(1,7 TO 8)="Ex": LET
l$(1,1 TO 6)=STR$ s: LET l$(2,1
TO 6)=STR$ k: LET l$(2,7 TO 8)="
Ey": PRINT  PAPER 5; INK 0;AT 13
,0;l$(1);AT 14,0;l$(2)
1900 LET l$(3,1 TO 4)=STR$ VAL r
$: LET l$(3,5 TO 8)="*ANG": PRIN
T  PAPER 2; INK 7;AT 15,0;l$(3)
1910 LET l$(9,1 TO 6)=STR$ i: LE
T l$(9,7 TO 8)="Tx": LET l$(10,1
 TO 6)=STR$ j: LET l$(10,7 TO 8)
="Ty": PRINT  PAPER 6; INK 1;AT
```

duzca primero –6.4 y después 6.4. Pulse ENTER y se ejecutará la orden de integración.

En matemáticas y física se utiliza muchas veces una función definida en un intervalo siguiente y así sucesivamente. Para este tipo de funciones sirve el comando ((W)).

Si pulsa ((W)) le pedirá una función que tendrá que introducir manteniendo todas las normas explicadas para este caso y, posteriormente, le preguntará "(HASTA?) dónde quiere realizar la representación" con esa función. Si x, todavía es menor que Ex, le volverá a pedir otra y hasta dónde debe estudiarla, y así hasta que X > = Ex, en cuyo caso termina.

Si en un momento dado desea anular la orden de integración, traslaciones, gráfica por intervalos, rotación o reponer la función, si estuviera elegida la inversa, pulse ((0)).

Las dos próximas órdenes nada tienen que ver con la representación gráfica de las funciones y sí con sus valores. Supongamos que desea calcular la imagen de un número real o de un conjunto de ellos sin tener que hacer una representación. Para el primer caso tenemos ((K)) y para el segundo ((U)). En ((K)) le pedirá la "ABSCISA X" y, después de pulsar ENTER, le escribirá en la parte superior de la pantalla su F(X). En ((U)), le pedirá el incremento de x y aparecerá una tabla de valores en la que se incluye: X, F(x) y F'(x), dando un sonido grave si la función es decreciente y agudo para creciente.

Hay veces que se quiere estudiar como varían los valores de F(x) cuando varían algunos de los coeficientes que la componen. Por ejemplo:

SEN x, SEN (2*x), SEN (–2*x), etc. Para ello disponemos de la sentencia ((X)) en la que aparecen cinco parámetros: X(1), X(2), X(3), etc. a los que les daremos los valores que nos interese. Estos aparecerán en pantalla si pulsamos ((L)) y no son nulos los cinco parámetros. Para cambiarlos debe presionar, de nuevo, ((X)). Si en algunos de ellos pulsa ENTER, ese parámetro mantendrá el último valor introducido en él. Para hacerlos desaparecer basta con darles el valor cero.

Ejemplo: Pulse ((4)) e introduzca



x(1)*SEN (x(2)*x) en el numerador. DEN = ENTER. Ahora pulse ((X)) e introduzca sucesivamente: 2, 3, ENTER, ENTER, ENTER; con lo que habrá hecho X(1)=2, X(2)=3, X(3) = X(4) = X(5) = 0, o sea, estar estudiando:

2*SEN (3*x)

Utilice Ex = PI; Ey = 2.5

Por último, y no por ello menos importante, destacaremos lo que llamamos CURSOR PUNTUAL.

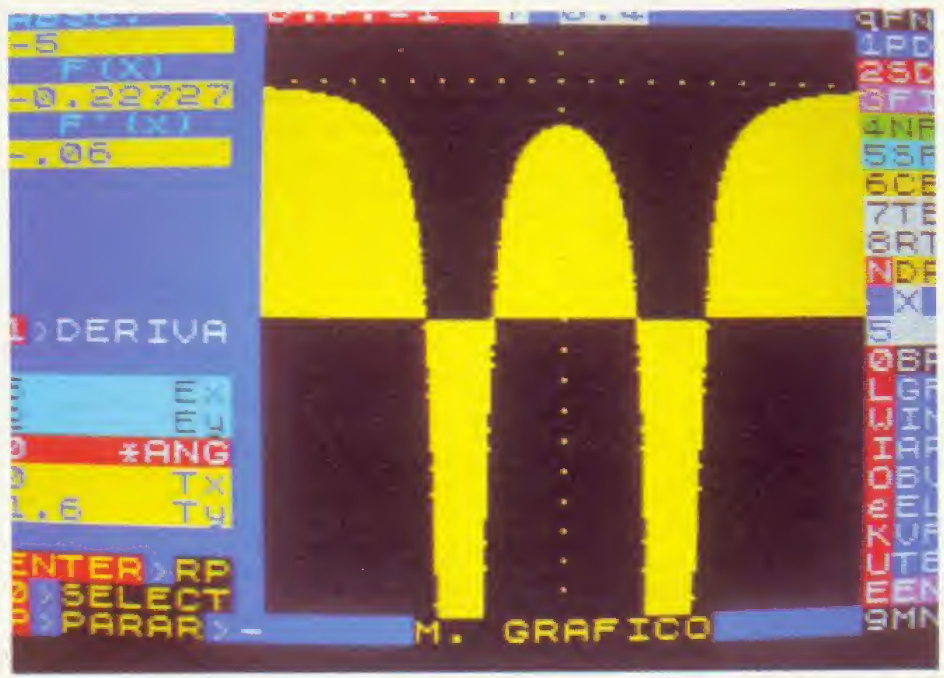
Si tenemos realizada la representación de F(x) ¿por qué no intentar resolver las ecuaciones: F(x) = 0; F'(x) = 0; F''(x) = 0? O si tenemos en pantalla dos funciones F(x)

```
16,0;1$(9);AT 17,0;1$(10)
1920 LET a=LEN (n$+d$+"1"): IF a
<29 THEN  INPUT "": PRINT #0; IN
VERSE 1;"Y="; INVERSE 0; PAPER 1
;n$;"/";d$
1930 LET p=0
1940 REM
1950 IF PEEK z=57 OR PEEK z=109
THEN  CLS : GO TO 2740
1960 LET qq=qq+.1: IF qq>15 THEN
 LET qq=0: INPUT "": PRINT #1:
INVERSE 0;"ENTER > GRAFICO ### 9
 > MENU"
1970 REM **SALIDA A M. GRAFICO *
1980 IF PEEK z<>13 THEN  GO TO 1
160
1990 LET ii=i: LET jj=j: IF c=0
AND cc=0 AND ii=0 AND jj=0 THEN
 GO TO 2020
2000 IF rf=0 AND dd<>3 THEN  GO
SUB 2130
2010 IF rf=1 OR dd=3 THEN  LET i
i=0: LET jj=0: GO SUB 2130
2020 IF f<>1 AND rp=0 THEN  PRIN
T AT 7,0;"          "
2030 IF f=1 THEN  LET ff=0
2040 LET a=LEN (n$+d$)+1: IF a<3
0 THEN  INPUT "": POKE z,64: PRI
NT #0; INVERSE 1;"Y="; INVERSE 0
; PAPER 1; INK 7;n$;"/";d$
2050 IF a>=30 THEN  INPUT "": PR
INT #0;" No presento F(x) "
2060 PRINT AT 19,0;"ENTER>RP";AT
 20,0;"O>SELECT";AT 21,0;"P>PARA
R>"
2070 IF rp=1 THEN  PRINT AT 19,0
;"        ";AT 21,0;"          "
2075 PRINT AT 21,14;"M. GRAFICO"
2080 IF gt=1 AND ggtt=1 THEN  PO
KE z,64: GO TO 550
2090 IF gt=1 AND ggtt=0 THEN  LE
T ggtt=1: GO TO 520
2100 POKE z,64
2110 BEEP .05,0: GO TO 510
2120 REM **** TRASL EJES ****
2130 FOR q=1 TO 19
2140 IF ABS c<s1 THEN  PLOT  OVE
R 1;152+c*80/s1,7+8*q
2150 IF ABS cc<k1 THEN  PLOT  OV
ER 1;72+8*q,87+cc*80/k1
2160 NEXT q
2170 FOR q=1 TO 19
2180 IF ABS ii<s THEN  PLOT 152+
ii*80/s,7+8*q
2190 IF ABS jj<k THEN  PLOT 72+8
*q,87+jj*80/k
2200 NEXT q: IF ABS c<s1 AND INT
 (c*10/s1)=c*10/s1 AND ABS cc<k1
 AND INT (cc*10/k1)=cc*10/k1 AND
 INT (c*100/s1)<>INT (ii*100/s)
AND INT (cc*100/k1)<>INT (jj*100
/k) THEN  PLOT  OVER 1;152+c*80/
s1,87+cc*80/k1
2210 LET s1=s: LET k1=k: LET c=i
i: LET cc=jj
2220 RETURN
2230 REM ****  CURSOR  *****
2240 LET zz=0: LET vv=0
2250 PRINT AT 0,0; INK 5;"ABSC.
 X";AT 2,0;"ORDE.  Y"
2260 IF CODE INKEY$=56 THEN  LET
 zz=zz+1: IF zz>=79 THEN  LET zz
=79
2270 IF CODE INKEY$=55 THEN  LET
 vv=vv+1: IF vv>=79 THEN  LET vv
=79
2280 IF CODE INKEY$=54 THEN  LET
 vv=vv-1: IF vv<=-79 THEN  LET v
v=-79
2290 IF CODE INKEY$=53 THEN  LET
 zz=zz-1: IF zz<=-79 THEN  LET z
z=-79
2300 PLOT  INK 9; OVER 1; BRIGHT
 1;152+zz,87+vv: PAUSE 15: PLOT
 BRIGHT 1; OVER 1; INK 9;152+zz,
87+vv
2310 LET p$(1)=STR$ (INT ((zz*s/
80-i)*1E6+.5)/1E6): LET p$(2)=ST
R$ (INT ((vv*k/80-j)*1E6+.5)/1E6
): PRINT AT 1,0;p$(1);AT 3,0;p$(
2)
2320 IF INKEY$="c" THEN  PRINT A
T 21,14;"           ";AT 1,0;"
```



y G(x) ¿por qué no resolver la ecuación F(x) = G(x)? Esto se puede conseguir mediante el uso de ((C)). Pulsando esta tecla, aparecerá en el centro de la pantalla un punto que puede ser desplazado con ((5)), ((6)), ((7)) y ((8)). Las coordenadas de ese punto aparecen en pantalla y, por tanto, si situamos el punto sobre la intersección de F(x) con el eje OX, tendremos la solución de F(x) = 0 y si lo hacemos sobre la función G(x), obtendremos el punto de corte de ambas funciones. Es muy práctico para saber las coordenadas de un punto de la pantalla. Si quiere deshacerse de él, pulse de nuevo ((C)).

Hay veces, cuando la función es muy larga, que al introducirla se sube a la pantalla. Esto ocurre sobre todo con las funciones SEN X, COS X, LN X, etc. (siempre que sean muy largas). Esto es debido a que el ordenador cuenta las letras de la cadena NUM/DEN y, si es menor que 29, la presenta en la parte inferior de la pantalla. En caso contrario no lo hace, pero el ordenador cuenta SEN X como dos letras y no cinco (contando el espacio). En este caso introduzca, por ejemplo, en el numerador, +0+0+... para sobrepasar las 29 le-




```
     ";AT 3,0;"           ": POKE z
,64: LET qq=0: LET p=1: GO TO 11
60
2330 GO TO 2260
2340 REM **** CONICAS ****
2350 FOR x=1 TO 5: PRINT AT 3+x,
0;"        ": NEXT x
2360 LET a$(1)=STR$ b: PRINT AT
0,9;"T.P.="; PAPER 2;a$(1)
2365 PRINT AT 21,14;"M. GRAFICO"
2370 PRINT AT 0,0; INK 5;"ABSC.
 X";AT 2,0;"ORDE.  Y"
2380 IF dd=1 OR dd=2 THEN  PRINT
 AT 4,0;"          "
2390 IF f=1 THEN  LET f=0: LET f
f=0: PRINT AT 7,0;"          ";AT
8,0;"          "
2400 IF gt=1 THEN  PRINT AT 10,0
;"          "
2410 PRINT AT 21,0; FLASH 1; INV
ERSE 1;"SEMIEJEX": INPUT sm: PRI
NT AT 21,0; FLASH 1; INVERSE 1;"
SEMIEJEY": INPUT sn: POKE z,64:
PRINT AT 21,0;"          "
2420 IF sm=sn THEN  PRINT AT 11,
0;"*CIRCULO"
2430 LET ii=0: LET jj=0: GO SUB
2130
2440 IF sm<>sn THEN  INPUT "": P
RINT #0; INVERSE 1;" a= ";sm;" *
** ";"b= ";sn;" ": POKE z,64
2450 IF sm=sn THEN  INPUT "": PR
INT #0; INVERSE 1;" R= ";sm;" "
2460 BEEP .05,0
2470 IF sm<>sn AND s=k THEN  LET
 l$(8,7 TO 8)="*e": LET l$(8,1 T
O 6)=STR$ (INT ((SQR ABS (sm*sm-
sn*sn)/sm)*1E6+.5)/1E6): PRINT A
T 12,0; PAPER 4; INK 0;l$(8)
2480 IF sm=sn AND s=k THEN  LET
l$(8,7 TO 8)="*A": LET l$(8,1 TO
 6)=STR$ (PI*sm*sm): PRINT AT 12
,0; PAPER 6; INK 0;l$(8)
2490 LET e$(2)=STR$ s: LET e$(1)
=STR$ k: PRINT AT 0,19;e$(1);AT
11,29;e$(2)
2500 PRINT AT 19,0;"ENTER>RP";AT
 20,0;"0>SELECT";AT 21,0;"P>PARA
R>"
2510 FOR n=0 TO 2*PI STEP 2*PI/b
2520 LET x=sm*COS n+i: LET y=sn*
SIN n+j: IF rr<>0 THEN  LET x1=(
x-i)*COS rr-(y-j)*SIN rr: LET x2
=(x-i)*SIN rr+(y-j)*COS rr: IF A
BS (x1+i)<s AND ABS (x2+j)<k THE
N  PLOT 152+(x1+i)*80/s,87+(x2+j
)*80/k
2530 IF rr=0 AND ABS x<s AND ABS
 y<k THEN  PLOT 152+x*80/s,87+y*
80/k
2540 IF rr<>0 THEN  LET p$(1)=ST
R$ (INT ((x1+i)*1E6+.5)/1E6): LE
T p$(2)=STR$ (INT ((x2+j)*1E6+.5
)/1E6): PRINT AT 1,0;p$(1);AT 3,
0;p$(2)
2550 IF rr=0 THEN  LET p$(1)=STR
$ (INT (x*1E6+.5)/1E6): LET p$(2
)=STR$ (INT (y*1E6+.5)/1E6): PRI
NT AT 1,0;p$(1);AT 3,0;p$(2)
2560 IF PEEK z=48 THEN  LET b=2:
 POKE z,64: PRINT AT 12,0;"
   ": GO TO 780
2570 IF PEEK z=112 OR PEEK z=80
THEN  PAUSE 0
2580 IF PEEK z=13 THEN  PRINT AT
 0,18;"E": LET n=n+2*PI/b
2590 IF PEEK z<>13 THEN  PRINT A
T 0,18;" ":
2600 NEXT n
2610 PRINT AT 12,0;"          ";AT
 0,18;" ";AT 1,0;"          ";AT 3
,0;"          ";AT 19,0;"          "
;AT 20,0;"          ";AT 21,0;"
      "
2620 POKE z,64: LET b=2: LET p=1
: GO TO 780
2630 REM ******  EJES CART. ***
2640 PRINT AT 21,9;"
      "
2650 FOR n=1 TO 10: PRINT AT n,9
; BRIGHT 1; PAPER pa;"
          ";AT (21-n),9;"
          ": NEXT n
2660 FOR n=1 TO 19: PLOT 72+8*n,
87: PLOT 152,7+8*n: NEXT n
2670 PRINT  INVERSE 1;AT 0,17;"Y
```

tras. Esto es poco probable que ocurra.

Este programa puede parecer largo para teclear pero, como han visto, sus aplicaciones son importantes.

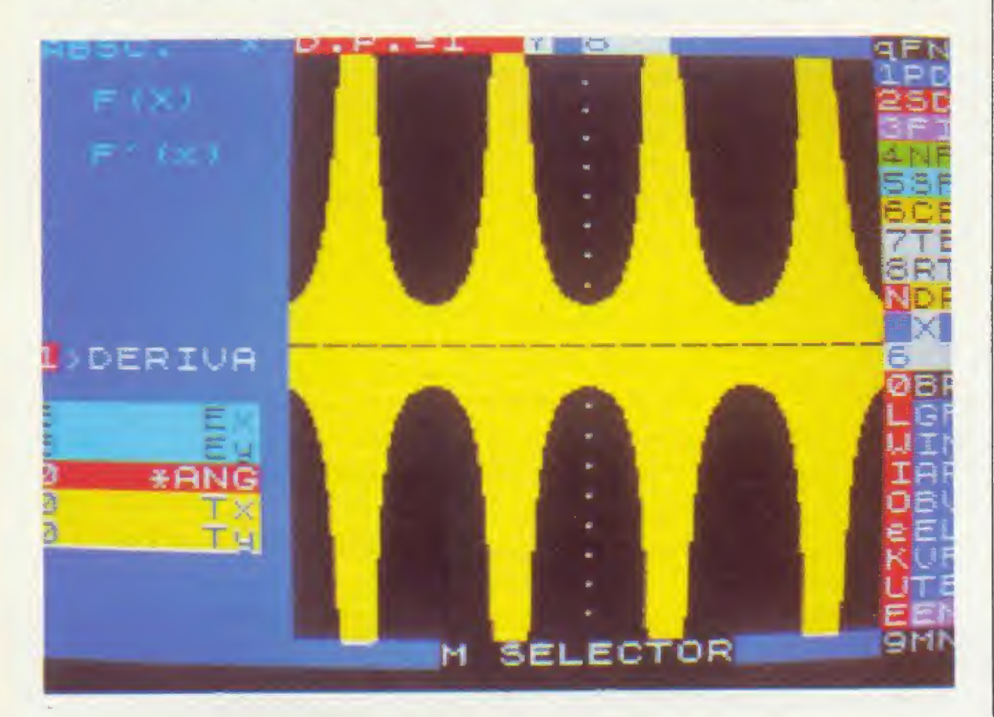
A los estudiantes aficionados a la "masacre" de marcianitos les diría que ellos tienen el mismo derecho a la vida que nosotros y que un micro-ordenador también sirve para otras cosas menos agresivas y, seguramente, más instructivas.

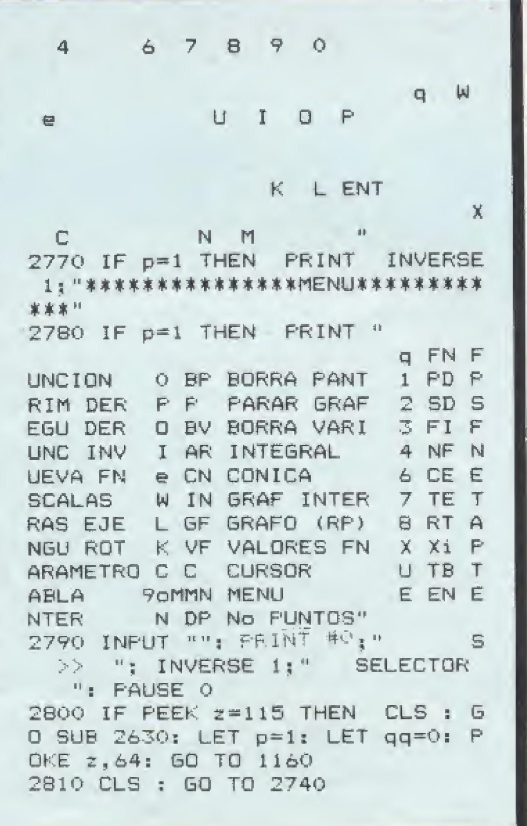
Para practicar, utilice las funciones y derivadas, que se hallan en el cuadro de la pág. 10.

Andrés Sánchez



José Villena, Félix Peral y Carmina Pérez en una de las clases de matemáticas.





```
"; PRINT  INVERSE 1;AT 10,30;"X"
2680 PRINT AT 0,29; PAPER 0;"qFN
";AT 1,29;"1PD";AT 2,29;"2SD";AT
 3,29;"3FI";AT 4,29; INK 0;"4NF"
;AT 5,29;"       6CE";AT 6,29; PA
PER 7;"7TE";AT 7,29;"8RT";AT 8,2
9; INK 7;"N"; INK 0;"
DP";AT 9,29;"XXi";AT 21,29; INK
7;"9MN"
2690 PRINT AT 12,29;"OBP";AT 13,
29;"LGF";AT 14,29;"WIN";AT 15,29
;"IAR";AT 16,29;"OBV";AT 17,29;"
eCN";AT 18,29;"KVF";AT 20,29;"EE
N";AT 19,29;"UTB"
2700 RETURN
2710 REM  ** DEF  VERIABLES **
2720 LET rf=0: LET ee=0: LET rp=
0: LET c=0: LET cc=0: LET ii=0:
LET jj=0: LET cg=0: LET z$="0":
LET x$="1": LET c$="0": LET v$="
1": LET ss=0: LET x2=0: LET x3=0
: LET b=2: LET f=0: LET ff=0: LE
T x1=80: LET ggtt=0: LET gt=0: L
ET r$="0": LET p=0: LET pp=0: LE
T z=23560: LET j=0: LET rr=0: LE
T qq=0: LET e=0: LET dd=0: LET i
=0: LET a=1: LET h=.001
2730 POKE 23658,0: INK 7: BORDER
 1: PAPER 1: CLS : CLS
2740 REM     ***** MENU  *****
2750 LET p=1
2760 IF p=1 THEN  PRINT "1  2  3
   4     6  7  8  9  0

                              q  W
  e              U  I  O  P



                         K  L ENT
                                  X
  C            N  M          "
2770 IF p=1 THEN  PRINT  INVERSE
 1;"****************MENU**********
***"
2780 IF p=1 THEN  PRINT "
                              q FN F
UNCION    O BP BORRA PANT  1 PD P
RIM DER   P P  PARAR GRAF  2 SD S
EGU DER   O BV BORRA VARI  3 FI F
UNC INV   I AR INTEGRAL    4 NF N
UEVA FN   e CN CONICA      6 CE E
SCALAS    W IN GRAF INTER  7 TE T
RAS EJE   L GF GRAFO (RP)  8 RT A
NGU ROT   K VF VALORES FN  X Xi P
ARAMETRO C C  CURSOR       U TB T
ABLA     9oMMN MENU        E EN E
NTER      N DP No PUNTOS"
2790 INPUT "": PRINT #0;"      S
 >>  "; INVERSE 1;"   SELECTOR
 ": PAUSE 0
2800 IF PEEK z=115 THEN  CLS : G
O SUB 2630: LET p=1: LET qq=0: P
OKE z,64: GO TO 1160
2810 CLS : GO TO 2740
```

Al acabar las clases es cuando comienzan las "otras" clases. Para poder alcanzar el aprobado en unos casos, o para mejorar los conocimientos en otros, al finalizar sus clases de BUP o COU, pequeños grupos de estudiantes se disponen cada día a aprender algo más de matemáticas, ayudados por un buen programa de ordenador.

Cabe preguntarse si no se corre el riesgo de olvidar la forma de resolución de una función. «El ordenador es una buenísima pizarra» comenta José Villena. «Tiene más retentiva que las fotos o la pizarra que nunca pueden dar la idea de dinamismo».

Con mucho, el más satisfecho es Andrés, profesor y autor del programa: «En el Spectrum tenemos un pequeño laboratorio. Puedes ver los efectos, variando distintos parámetros, como en el típico problema de la polea, donde al cambiar las masas puedes ver claramente la aceleración que se produce. Especialmente en el campo de la física existen muchos conceptos abstractos sobre los que el ordenador tiene mucho que decir. Hay algunos programas de trigonometría, tiro oblicuo, ondas, pero nada sobre matemáticas con una mínima seriedad. La mayoría de los programas didácticos sobre matemáticas presentan el inconveniente de no contar con que el denominador de una ecuación puede valer cero, por lo que son frecuentes los errores a pesar de ser un tema trivial».

Todos los caminos conducen a ROM

La instrucción RST 56 (38H) la utiliza el Spectrum para analizar el teclado y actualizar la variable del sistema FRAMES. Se le llama mediante la rutina de enmascaramiento. Si todo lo que necesitamos es comprobar si se ha pulsado una tecla, bastará con analizar el bit 5 de la variable situada en la dirección 23611. Si está a 1 entonces se ha pulsado una tecla.

Observe que esto hay que volver a inicializarlo. El código para la última tecla pulsada se encuentra a partir de la 23560. Con frecuencia será suficiente mientras se realiza la entrada de datos mediante el teclado pero tiene sus desventajas. Primero, sólo se actualiza 50 veces por segundo (60 en América). De manera que no se puede efectuar una RES 5 inmediatamente seguido de una comprobación del bit 5 de 23611, ya que debido a la interrupción, el teclado no se habría comprobado. Por otro lado, si la interrupción no está conectada, el teclado no se comprobará nunca hasta que lo intente. Pruebe a ejecutar lo siguiente:

```
LD      HL, 23611
RES     5,(HL)
LD      A, FF
LD      (23552), A
RST     56
XOR     A
BIT     5, (HL)
JP      Z, TECLAS
LD      A, (23560)
TECLAS  (cualquier cosa)
```

Esto devuelve el código de la tecla pulsada en el párrafo detallado con TECLAS y lo sitúa en el registro A, o pone O si no se ha pulsado una tecla.

En la rutina anterior la instrucción LD HL, 23611 se podía haber omitido y haber probado el registro (IY+1) ya que el IY contiene la dirección de la variable del sistema ERR NE (23610) y se utiliza por la ROM para direccionar las variables del sistema. La razón es la siguiente, si se utiliza el registro IY en los programas, hay que asegurarse de que las interrupciones esten desconectadas o dirigidas hacia nuestras rutinas. Habrá que reinicializar el registro IY a la dirección correcta antes de utilizar la ROM o volver al modo normal de interrupciones.

El segundo problema está en que sólo se puede comprobar si se ha pulsado una tecla, no varias.

Para solucionar esto hay que escribir una rutina propia para la comprobación del teclado. Si desea saber si se ha pulsado una tecla sin especificar o varias, tendrá que realizar algo parecido a lo siguiente:

```
XOR     A (asegurarse de que A = O)
IN      A, (FEH); (FEH es el port de
        entrada desde el teclado etc.)
LD      D, 31
AND     D; (elimina etc de los 3 bits
        más altos).
XOR     D
JR      Z, TECLAS
```

Si cargamos en el registro A el valor de la línea a comprobar antes de la instrucción IN A, (FEH), podremos comprobar adecuadamente si se ha pulsado una tecla.

El teclado del Spectrum esta dispuesto de la manera siguiente. El primer caracter en cada caso es BIT O y en el segundo BIT 4, el número hexadecimal se ha de cargar en el registro A para poner el Bit de la línea que se comprueba.

CAPS SHIFT-V	FEH
A - G	FDH
Q - T	FBH
1 - 5	F7H
0 - 6	EFH
P - Y	DFH
ENTER - H	BFH
SPACE - B	7FH

De manera, que para saber si se ha pulsado ENTER el programa sería:

```
LD      A, BFH
IN      A, (FEH)
AND     1
JR      Z, ENTER PULSADO
```

Para comprobar si se ha pulsado más de una tecla se pueden utilizar los operadores lógicos AND, OR, etc., si se encuentran en la misma línea.

COPIA DE LA PANTALLA A LA IMPRESORA: CALL 3756 (EACH)–

Esta rutina no requiere una preparación previa y una llamada directa (CALL) permitirá copiar la pantalla a la ZX printer.

LISTANDO LOS GRAFICOS A LA IMPRESORA: CALL 3789 (EDCH)–

Esta es similar a la rutina anterior en cuanto a que utiliza el buffer de la impresora y vuelca su contenido a ella. Se utiliza por la rutina RST 16, que normalmente trata a los buffers como líneas de pantalla (con una altura de 8 pixels). Situando los gráficos en el buffer una línea a la vez y a continuación se hace una llamada (CALL) a esta rutina la impresora copiará el buffer.

Observe que, sin embargo, la salida es de 32 bytes por línea de pixels con la siguiente línea de pixels inmediatamente después y no como en la pantalla. El buffer se pone a cero después de la llamada (CALL).

LIMPIANDO EL BUFFER DE LA IMPRESORA: CALL 3807 (EDFH)–

Sólo limpia el buffer, poniendo ceros.

UTILIZANDO BEEP: CALL 949 (3B5H)–

El par de registros DE almacenan la longitud de salida y el par de registros HL la frecuencia. O es para alta frecuencia y FFFFH es para baja frecuencia. El problema de esta rutina es que la duración de esta rutina depende de la frecuencia. De forma que si la frecuencia se corta se dobla la duración para un valor dado en DE.

Los valores actuales se calculan de la forma siguiente:

HL = (437,500/frecuencia) - 30.125

DE = duración * frecuencia
La duración ha de estar en segundos.

La razón de restar 30.125 del registro HL es porque la rutina tarda 120.5 T estados para generar la nota y actualizar su registro.

La C es 261 Hz, aproximadamente, de forma que el valor de HL ha de ser sobre 1646 en decimal y DE para un segundo valdrá 261 en decimal.

Recuerde que las interrupciones desde la ULA ocurren 50 veces por segundo y que degenerarán la salida del sonido si la rutina se halla en los 16K inferiores de la RAM.



Y es lo lógico. Si has elegido el mejor microordenador del mercado, no vas a repararlo con cualquiera.

Solo HISSA, Servicio Oficial de Investrónica para los ordenadores SINCLAIR, te puede garantizar la utilización de piezas originales SINCLAIR y expertos técnicos en reparación.

Y recuerda que no tendrás sobresaltos con el precio.

"COSTE ESTANDAR POR REPARACION"

ZX 81:	3.150 Ptas.
Spectrum 16K:	5.250 Ptas.
Spectrum 48K:	6.300 Ptas.

Acude a la delegación **HISSA** más cercana.

DELEGACIONES HISSA

C/. Aribau, n.º 80, piso 5.º 1.º
Telfs.: (93) 323 41 65 - 323 44 04
08036 BARCELONA

P.º de Ronda, n.º 82, 1.º E
Telf.: (958) 26 15 94
18006 GRANADA

C/. Universidad, n.º 4 - 2.º 1.º
Telf.: (96) 352 48 82
46002 VALENCIA

C/. San Sotero, n.º 3
Telfs.: 754 31 97 - 754 32 34
28037 MADRID

C/. 19 de Julio, n.º 10 - 2.º local 3
Telf.: (985) 21 88 95
33002 OVIEDO

Avda. de Gasteiz, n.º 19 A - 1.º D
Telf.: (945) 22 52 05
01008 VITORIA

C/. Atares, n.º 4 - 5.º D
Telf.: (976) 22 47 09
50003 ZARAGOZA

C/. Avda. de la Libertad, n.º 6. Bloq. 1.º Entl. Izq. D.
Telf. (968) 23 18 34
30009 MURCIA

C/. Hermanos del Río Rodríguez, n.º 7 bis
Telf.: (954) 36 17 08
41009 SEVILLA

C/. Travesía de Vigo, n.º 32 - 1.º
Telf. (986) 37 78 87
6 VIGO

GHOSTBUSTERS

Distribuidor: Proeinsa
Formato: *cassette*
Memoria requerida: 48K



Plano de la ciudad, recorrido incesantemente por fantasmas en dirección al edificio de Zuul.



Morada de Zuul. Los cazafantasmas se preparan para atrapar al fantasma con su rayo especial.

No hace falta decir que este juego está inspirado en la película del mismo nombre. Lo que se busca es la caza de fantasmas. Para ello, al igual que los héroes de la pantalla grande, hay que partir por equiparse bien. Lo primero es *seleccionar el vehículo.* Se puede elegir un compacto, un coche fúnebre, una camioneta rural o uno muy especial, con carga para 7 equipos y 160 millas/hora. La elección no es sólo cuestión de gusto, sino principalmente de dinero. Una vez elegido el coche hay que equiparlo convenientemente según el dinero que le haya quedado.

En esta segunda fase de *selección de equipos* dispone de equipos de monitores y detección (detector de energía espectral, intensificador de imágenes...), de captura (carnada para fantasmas, trampas...) y de almacenaje (rayo láser...). Armado hasta los dientes puede pasar a la tercera fase: El *plano de la ciudad,* el horripilante templo de Zuul, el cuartel central y los edificios adyacentes aparecen en la pantalla, rápidamente recorridos por las ígnotas criaturas.

Con el vehículo perfectamente equipado su misión es llegar los antes posible a la calle donde se producen los disturbios. Una vez allí comienza la cuarta y más delicada fase: *Atrapar al fantasma.* Para ello se deposita la trampa en el centro del edificio y se colocan dos *ghostbuster* a cada lado del edificio al objeto de disparar sus rayos sin que estos se crucen. Si logra atraparlo, ya estará cerca de su objetivo final: *El templo de Zuul* y el terrible monstruo de caramelo.

Control: joystick, teclado.
Jugadores: 1
Gráficos: Diversas pantallas con alta definición de caracteres.
Sonido: música de la versión cinematográfica.
Niveles de dificultad: no existen.
Originalidad: reproducción fiel de la versión cinematográfica, aprovechando las características del ordenador al máximo.
Conclusión: divertido y complejo programa. Habrá que esperar a ver si la versión del Spectrum es tan completa como ésta.

Aquí, a diferencia de la versión cinematográfica no hay que derrotar a Zuul, basta con poder introducir a los cazafantasmas en el templo, lo cual le costará lo suyo.

El programa es francamente bueno y los gráficos pueden producir la misma «alucinación» que ver un fantasma propiamente dicho. Solo un pequeño detalle: el programa analizado es la versión existente para el Commodore 64, motivo por el que no hemos entrado en detalle en la explicación de su funcionamiento. Para este mismo mes se espera la versión para el Spectrum de 48K con idénticas reglas de funcionamiento y gráficos. ¡Qué así sea! En la espera le aconsejamos se haga amigo de quien tenga un Commodore 64.

DECATHLON

Distribuidor: Zafiro
Formato: *cassette*
Memoria requerida: 48K



Todo está listo para empezar los 100 metros lisos. A esta prueba le siguen muchas otras.

Inspirado en los video-juegos de igual nombre, es uno de los programas más conocidos y que mayor expectación levanta por la utilización del teclado en lo que a movimiento del atleta se refiere.

Las competiciones, en un número total de 10, se reparten en las dos caras, denominadas DIA 1 y DIA 2. Las pruebas del primer día consisten en los 100 metros lisos, salto de longitud, lanzamiento de peso, salto de altura y los 400 metros lisos. El segundo día tiene igualmente cinco pruebas, pero su dificultad crece considerablemente. Se trata de los 110 metros con valla, salto de altura con pértiga, lanzamiento de disco, lanzamiento de jabalina y 1.500 metros.

Para el movimiento puede utilizarse el teclado del Spectrum o un *joystick*.

Al comenzar el juego se introducen las iniciales del "atleta de turno", con lo que se podrá competir posteriormente, dado que se guardan las puntuaciones y el marcador ofrece en todo momento las tres mejores puntuaciones para cada prueba.

Los gráficos son realmente excelentes. La pantalla se divide en tres partes: La parte superior contiene la puntuación donde se incluyen los puntos acumulados del atleta que está participando, la prueba que se está disputando y las tres mejores marcas de dicha prueba. Debajo de los marcadores se encuentra el público que lleno de júbilo se agita cada vez que se consigue entrar en una de las tres primeras puntuaciones y un poco más abajo el infatigable atleta que aguanta lo que le eche. Finalmente, en la parte inferior hay un indicador de velocidad y distintos tipos de información según la prueba a realizar (metros recorridos, ángulo de contacto, tiempo transcurrido...). Cuando falla en algún intento, generalmente por pisar una vez pasada la línea de salto, el atleta se rasca pensativamente la cabeza. En cambio cuando lo consigue, alza las manos en señal de victoria.

Otros detalles gráficos de gran vistosidad se dan en el salto de longitud, cuando un joven mide la distancia lograda, en el lanzamiento de disco, acompañado de su sombra o en los 100 metros lisos cuando el juez da la salida pistola en mano.

A estos efectos realmente logrados hay que añadir un buen acompañamiento musical entre prueba y prueba y, por supuesto, a la hora de subir al pódium.

Hay que destacar que no siempre salen todas las pruebas descritas, sino que el número depende de las puntuaciones obtenidas. A mayor destreza mayor número de pruebas o repetición de algunas y oportunidad de mejorar la puntuación. Motivo por el que no pudimos llegar a la prueba de los 1.500 metros por mucho que lo intentamos.

Desgraciadamente la distribuidora ha optado por ofrecer directamente la versión inglesa sin traducir, colocando la traducción de las instrucciones, la carátula de la cinta y un pequeño follèto en su interior explicando los mensajes que aparecen por pantalla. Unico aspecto negativo de este juego que exige serios entrenamientos para estar a la altura de los grades campeones.

No hay un número limitado de jugadores, por lo que cualquiera puede incorporarse a las olimpiadas, con el único *hándicap* de que el ordenador conserva los *records* y, por tanto, la competición se vuelva cada vez más dura y más emocionante.

Control: joystick, teclado.
Jugadores: 1, aunque el número de competidores es ilimitado.
Gráficos: excelentes.
Sonido: bueno.
Nivel de dificultad: dificultad progresiva según juego realizado.
Originalidad: versión del famoso video-juego.
Conclusión: altamente atractivo y de gran adición.

③

Descubrimiento de un nuevo lenguaje: PASCAL

En este tercer capítulo de la serie que venimos dedicando al PASCAL, vamos a tratar de explicar las sentencias de control de este lenguaje, que como veremos nos ofrece algunas ventajas respecto al BASIC, en este tipo de instrucciones.

Las sentencias de control dentro de un programa nos dan la posibilidad de elegir entre dos o más alternativas (bifurcaciones), y además el repetir un conjunto de instrucciones tantas veces como sea necesario. Estas repeticiones son los llamados bucles. Algo importante a tener en cuenta es no caer nunca en un bucle infinito. Por tanto, siempre dentro de un bucle, habremos de evaluar una condición para que éste no se ejecute indefinidamente. Una vez hecha esta pequeña introducción a las sentencias de control, vamos a empezar a explicar detalladamente cada una de ellas.

Sentencia IF

La sentencia condicional IF se utiliza de la misma forma que en BASIC, y es la que nos permite tomar una de las dos alternativas posibles de la condición puesta.

Siguiendo el diagrama sintáctico de la figura 1, si la evaluación de la condición es cierta, se ejecutará la parte del programa que sigue a THEN, es decir la sentencia 1 (la cual puede ser una sentencia compuesta), y en el caso de que la evaluación de la condición sea falsa se pasa a ejecutar la parte del programa que sigue a ELSE, que como vemos es la sentencia 2.

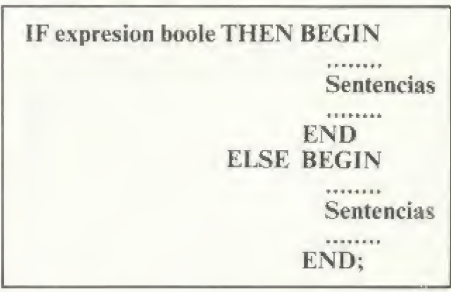
Un ejemplo sencillo de esta instrucción podría ser:

```
VAR
    Número: integer;
BEGIN
    ........
    IF Numero < 10 THEN WRITE
                  (NUMERO)
                  ELSE READ
                  (NUMERO);
    ........
END.
```

En este ejemplo, siempre que un número leído fuera menor que 10 se escribiría (parte correspondiente a THEN), mientras que si fuera mayor que 10 se ignoraría y se pasaría directamente a leer otro. Los puntos suspensivos significan que puede haber instrucciones en su lugar. El funcionamiento de esta sentencia mediante un organigrama se ilustra en la figura 2.

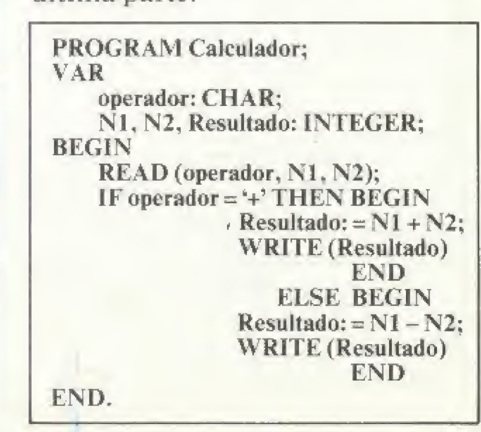
Por supuesto que esta condición dentro de un programa será siempre una expresión de BOOLE, que ya todos conocemos. Si nos fijamos en el diagrama sintáctico vemos que la sentencia IF puede o no llevar ELSE. Si no existiera la parte correspondiente a ELSE, y la evaluación de la condición fuera falsa, el programa se seguiría ejecutando secuencialmente (la siguiente sentencia a IF).

Cuando la sentencia alternativa esté compuesta por un conjunto de sentencias, éstas deberán agruparse entre las palabras BEGIN END para así formar un bloque. Por tanto, el caso más general en el que tanto la sentencia 1 como la sentencia 2 son compuestas será:



```
IF expresion boole THEN BEGIN
                        ........
                        Sentencias
                        ........
                        END
                   ELSE BEGIN
                        ........
                        Sentencias
                        ........
                        END;
```

Vamos a ver un pequeño programa para comprender mejor esta última parte:



```
PROGRAM Calculador;
VAR
    operador: CHAR;
    N1, N2, Resultado: INTEGER;
BEGIN
    READ (operador, N1, N2);
    IF operador = '+' THEN BEGIN
                  Resultado: = N1 + N2;
                  WRITE (Resultado)
                       END
                  ELSE BEGIN
                  Resultado: = N1 – N2;
                  WRITE (Resultado)
                       END
END.
```

Fig. 1.

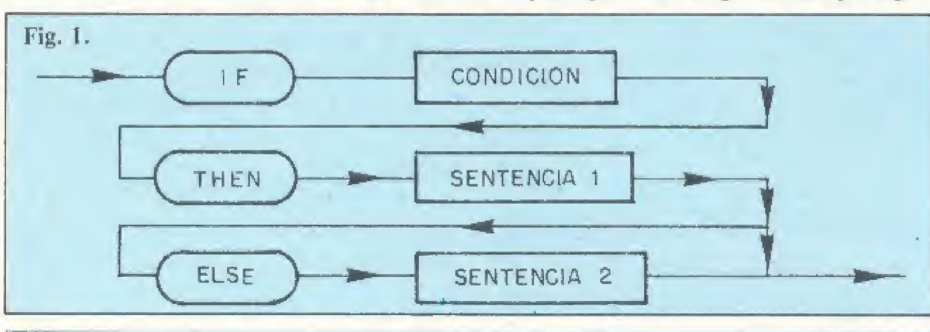
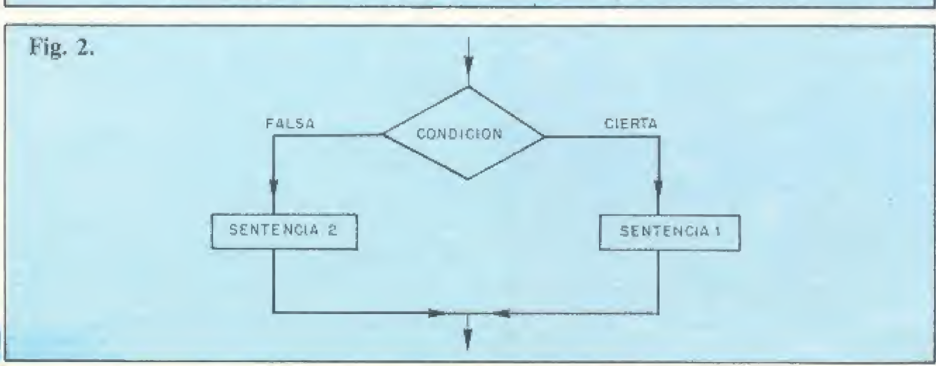


Fig. 2.



En este programa siempre que leamos el operador +, la condición del programa será cierta y elegiremos la alternativa THEN, calculándose la suma de los dos números.

En el caso de que el operador leído sea cualquier otro caracter, elegiremos la opción ELSE del IF, calculándose entonces la resta de ambos números. En ambos casos, ya que tanto el THEN como el ELSE se componen de más de una sentencia, sera necesario agrupar estas entre las palabras BEGIN-END.

Por último para terminar con esta sentencia condicional, indicaremos que pueden existir dentro de un programa una serie de sentencias IF anidadas. Vamos a explicarlo con un ejemplo:

```
IF condicion 1 THEN IF condicion
2 THEN sent 1 ELSE sent 2
```

De esta estructura se podrían dar dos interpretaciones:

a) El ELSE va con el primer IF
b) El ELSE va con el segundo IF

La interpretación válida será la segunda, puesto que el ELSE se corresponde siempre con el IF inmediatamente más cercano. Por tanto, dentro de una instrucción de este tipo se deberá cumplir la relación:

n.º IF $\geqslant$ n.º ELSE

La estructura de esta anidación de sentencias IF vista mediante un

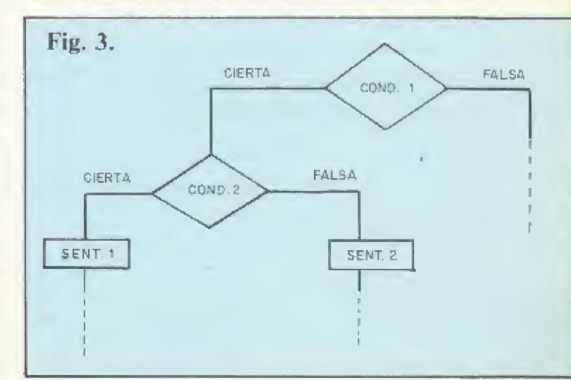


Fig. 3.

organigrama queda reflejada en la figura 3.

Para que no exista ambigüedad con la interpretación de estas instrucciones anidadas, se aconseja (aunque no es obligatorio), agrupar el IF "más interior" dentro de un bloque, es decir encerrarlo entre las palabras BEGIN-END. Así la expresión del ejemplo anterior quedaría:

```
IF cond1 THEN BEGIN IF cond 2 THEN sent 1 ELSE sent 2 END
```

En la que ya distinguimos perfectamente que el ELSE se corresponde con el segundo IF.

Vemos entonces que esta utilización de la instrucción IF anidadas nos permitirá mejorar los rendimientos de un programa.

Una vez vista la sentencia condicional vamos a pasar a ver las sentencias repetitivas. El PASCAL admite 3 clases de sentencias repetitivas: cuando se conoce el número de veces que se va a repetir un bucle se utilizará la sentencia FOR y cuando desconocemos de antemano las veces de repetición del bucle utilizaremos las instrucciones WHILE Y REPEAT.

Comenzaremos por WHILE.

Sentencia WHILE

Esta sentencia no tiene equivalente en BASIC, (ver figura 4). Para mostrar el funcionamiento de esta sentencia, utilizaremos el organigrama de la figura 5.

En esta sentencia se evalúa la condición, y si ésta no se cumple el bucle no se ejecuta. El bucle se ejecuta sólo si la condición sea cierta. Sabemos que el bucle de esta sentencia comienza detrás de la palabra DO, pero no podemos determinar de antemano donde finaliza. Por tanto, será necesario delimitar el bucle, lo cual conseguiremos como ya sabemos, encerrando éste entre las palabras BEGIN y END.

La forma de escritura entonces será:

```
WHILE condicion DO BEGIN
                        ........
                        SENTENCIAS
                        ........
                     END;
```

Por el contrario, si es una sola instrucción la que está incluida dentro del bucle, no será necesario encerrarla entre BEGIN-END, puesto que el punto y coma final de la sentencia indicará que ésta ya se ha acabado.

Un sencillo ejemplo para mostrar la escritura de una sentencia WHILE es:

```
........
........
WHILE numero > ∅ DO BEGIN
                     WRITE (numero);
                     READ (numero)
                  End;
```

Esta sentencia WHILE dentro de un programa nos servirá para escribir todos los números positivos que leamos. Si el número leído fuera negativo no se ejecutaría la sentencia ninguna vez. Para finalizar, es importante recordar que si no se cumple la condición inicial en esta instrucción, el bucle no se realizará ninguna vez.

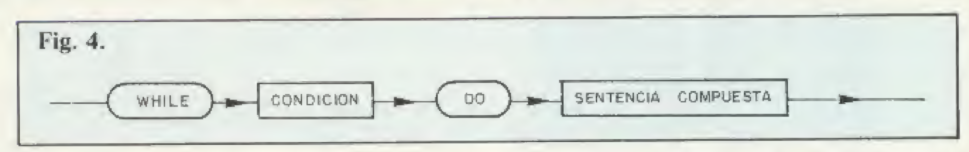


Fig. 4.

Sentencia REPEAT

Tampoco esta sentencia tiene equivalencia en BASIC, aunque por supuesto la podemos simular. Para seguir con el mismo orden que en la sentencia WHILE, el diagrama sintáctico correspondiente es el de la figura 6. La sentencia de control repetitiva de dicha figura, ejecuta al menos una vez el bucle correspondiente. (Para entender mejor el funcionamiento, ver la figura 7.) El conjunto de sentencias comprendidas entre las palabras REPEAT y UNTIL constituye el cuerpo del bucle de esta instrucción, teniendo en cuenta además que éste se ejecutará al menos una vez, ya que la comprobación de la condición es posterior a la ejecución del bucle.

Por tanto esta ejecución repetitiva se controla mediante la condición (expresión de Boole), la cual se evaluará después de cada iteración.

Deberemos tener en cuenta que dentro del conjunto de sentencias del bucle, habrá algo que haga va-

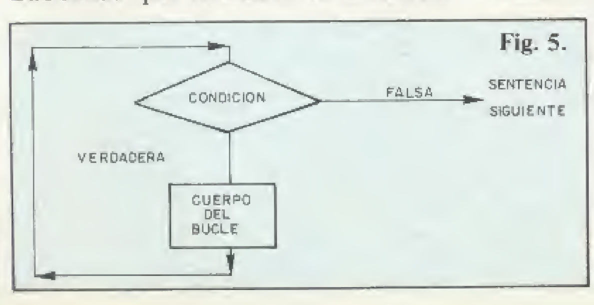


Fig. 5.

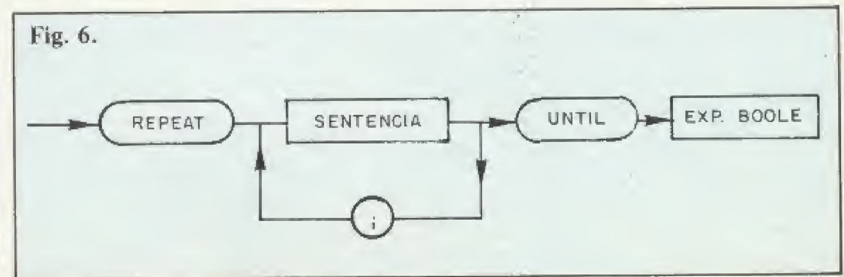


Fig. 6.


ESPECIFICAMENTE PARA QL
Mod. 1431 MZ 4
P.V.P. 74.500 Ptas.
INMEDIATAMENTE DISPONIBLE
ESPECIFICACIONES
MODELO:
CUB 1451/DQ3
14" QL MONITOR
Entrada RGB-TTL
Resolución (PIXELS)
653 (H) × 585 (V)
DOT PITCH 0,43 mm
Bandwidth 18 MHz
Específicamente diseñado para el QL que aprovecha su facilidad única de 85 columnas.
Mod. 1451/DQ3
P.V.P. 96.000 Ptas.
MICROVITEC 653
cub
COLOUR DISPLAYS
DISTRIBUIDO EN EXCLUSIVA POR: MULTILOGIC. Ramón de Santillán, n.° 15 28016 MADRID Tel. 458 74 75

riar la condición, para que este bucle no se convierta en infinito.

Un ejemplo típico para la compresión de la sentencia REPEAT, es hacer un programa que calcule la suma de N números.

```
PROGRAM serie:
VAR
      núm.; N : integer; I : integer;
      suma : real;
BEGIN
      READ (N);
      READ (num);
      I := 1; suma := 0;
      REPEAT
            suma := suma + num.;
            I := I + 1;
            READ (num.)
      UNTIL I = N;
      WRITE (suma)
END.
```

La variable N de este programa indicará el número de veces que se va a realizar el bucle, o sea, cuántos números vamos a sumar. Vemos en este problema que, a diferencia de la sentencia WHILE, aquí el bucle queda delimitado ya por la propia instrucción repetitiva, por las palabras REPEAT - UNTIL.

La condición viene impuesta por la expresión booleana I = N. Mientras que I no alcance el valor de N leído, el bucle se seguirá ejecutando. La instrucción que varía la condición dentro del bucle será I := I + 1, la cual conseguirá que nunca entremos dentro de un bucle infinito.

Vemos, por tanto, que la diferencia entre la sentencia WHILE y REPEAT será el lugar de evaluación de la condición. Mientras que en la sentencia WHILE la condición se evalúa antes y por ello puede que el bucle no se ejecute ninguna vez, en la sentencia REPEAT, la condición es posterior al bucle y éste se ejecutará al menos una vez.

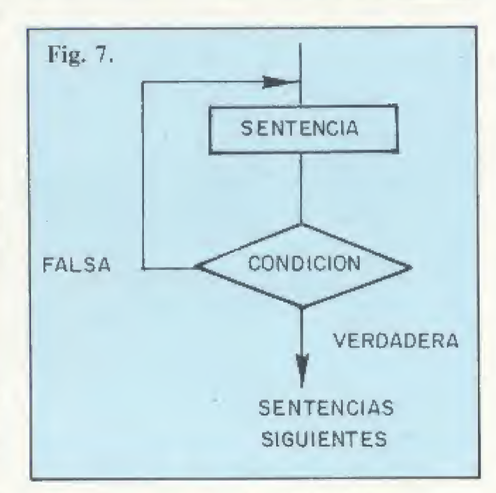


Fig. 7.

Sentencia FOR

Esta sentencia, equivalente a la sentencia FOR en BASIC, se utiliza cuando sabemos de antemano el número de veces que se va a realizar el bucle. Por tanto, esta sentencia estará controlada por un índice, el cual variará en sentido creciente o decreciente indicando el número de iteraciones del bucle. (Figura 8).

Deberemos tener en cuenta que tanto la variable de control, como el valor inicial y el valor final deberán ser del mismo tipo ordinal, y no deben alterarse por el usuario en la ejecución de esta sentencia.

La opción TO del FOR contará ascendentemente, mientras que la opción DOWNTO lo hará descendentemente.

Si el valor inicial de la variable es mayor que el valor final, el bucle FOR no se ejecutará ninguna vez.

El funcionamiento de la sentencia FOR representado mediante un organigrama se ilustra en la figura 9.

Como siempre, si el bucle está compuesto por más de una sentencia deberemos delimitarlo, usando como hasta ahora las palabras BEGIN y END.

Deberemos tener en cuenta que a un bucle FOR sólo se puede acceder por el principio, pero es posible la salida de un bucle para ejecutar alguna otra parte del programa.

Las sentencias FOR pueden estar anidadas, con la única restricción de no mezclar los bucles. Algunas estructuras válidas de las sentencias FOR anidadas pueden verse en las figuras 10 y 11.

Una estructura no válida, ya que se mezclan dos bucles podría ser la de la figura 12.

Dentro de un programa con bucles anidados, siempre se ejecutará ante el FOR más interior, y en el

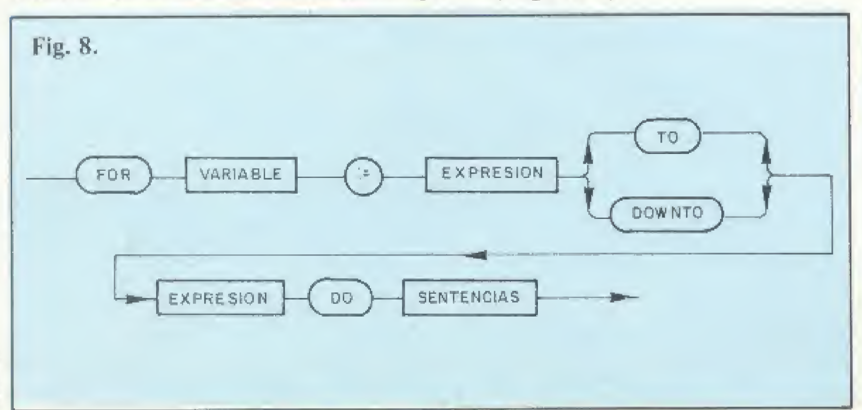


Fig. 8.

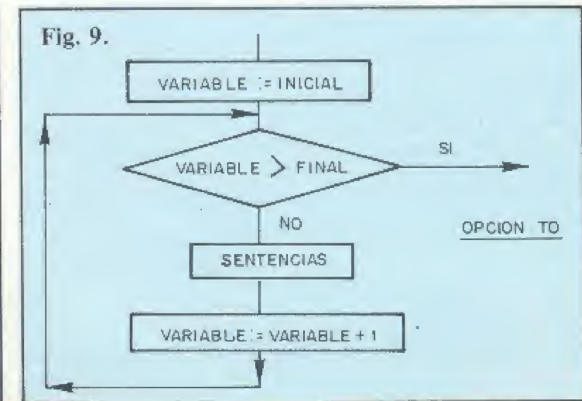
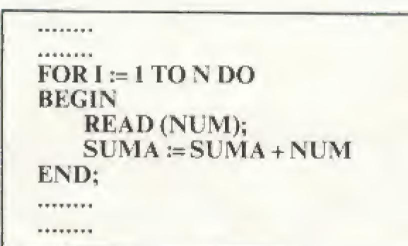


Fig. 9.

caso de que haya dos FOR al mismo nivel se ejecutará antes el primero que aparezca.

El programa que calcula la suma de N números hecho con sentencias FOR sería:



```
........
........
FOR I := 1 TO N DO
BEGIN
     READ (NUM);
     SUMA := SUMA + NUM
END;
........
........
```

En el próximo capítulo veremos un sencillo programa de conversión de números árabes a romanos, en el que se emplean con bastante frecuencia estas sentencias de control.

```
BEGIN
        BEGIN

        END
        BEGIN

        END
END
```

Fig. 10.

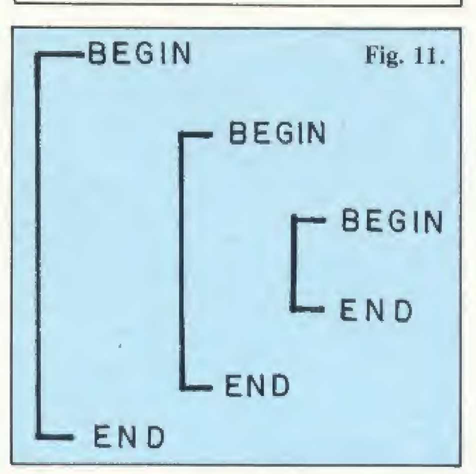


Fig. 11.

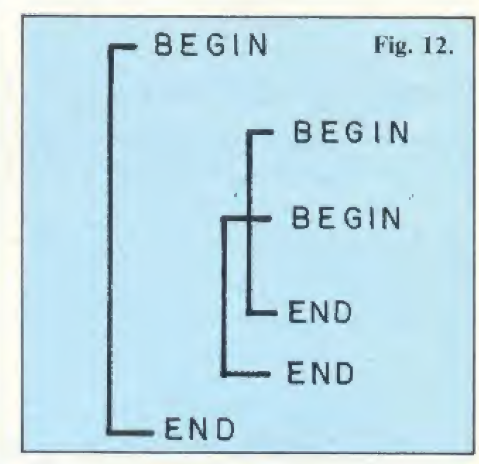


Fig. 12.

Lápiz óptico

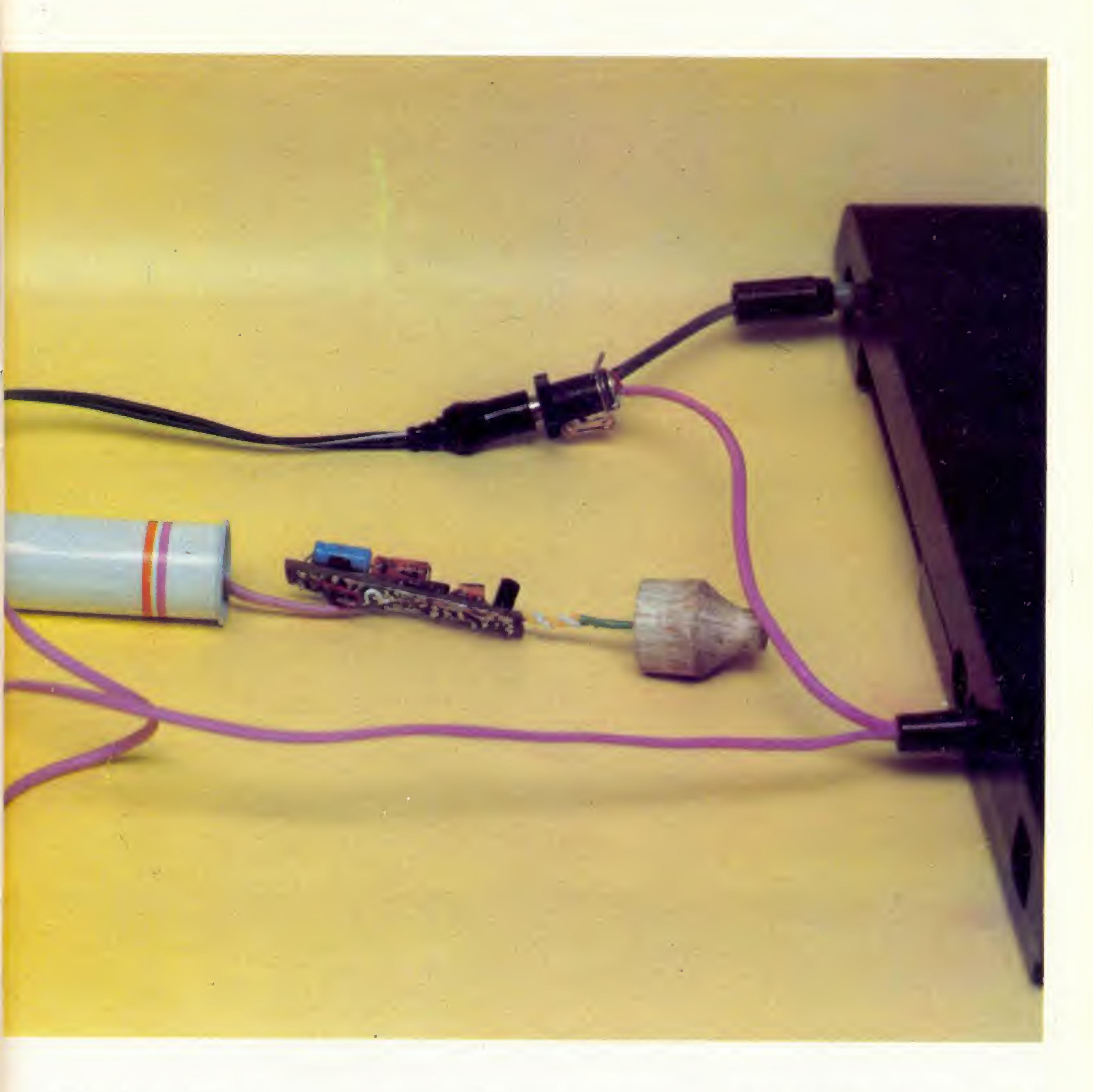
El sueño de Sinclair ha sido siempre fabricar equipos baratos, y podemos decir que se ha convertido en realidad en reiteradas ocasiones, aunque dudamos que pudiera imaginar poder contar con un lápiz óptico por poco más de 500 ptas. El atractivo es doble para quienes además disfrutan con su realización.





Si alguien nos comentara la posibilidad de construir un lápiz óptico por menos de 500 ptas. y sin necesitar experiencia previa en montajes electrónicos, tal vez, no le creeríamos. Sin embargo, hay que confiar más en la gente, porque es CIERTO.

En este artículo, nos proponemos guiar la construcción de un sencillo —pero efectivo— lápiz



óptico. Se pretende que, incluso los menos experimentados, sean capaces de seleccionar menús, definir gráficos o escribir música sin tocar una tecla.

Podríamos decir que un lápiz óptico se diferencia del resto de los instrumentos en que si se apunta hacia un monitor o una pantalla de televisión, es capaz de detectar si se hace sobre una zona oscura o iluminada, e informar consecuentemente al ordenador con que está conectado. (Nótese que este efecto es raramente comprobable en una grapadora o un florero, por ejemplo).

¿Cómo funciona un lápiz óptico?

El funcionamiento de este extraño artilugio viene estrechamente ligado al modo de operar de la pantalla de televisión. Expliquemos muy brevemente y en forma aproximada, cómo se genera una imagen en la pantalla de nuestro aparato.

El tubo de televisión (fig. 1) dispone en su parte más estrecha de un dispositivo denominado cañón electrónico, que tiene la perversa misión de disparar un rayo de veloces electrones (haz electrónico)

cuya intensidad puede ser controlada.

A continuación, se dispone de unas bobinas deflectoras cuya misión es la de desviar a voluntad el estrecho haz. Existen pares de bobinas dispuestas horizontal y verticalmente. Actuando convenientemente sobre ellas, seremos capaces de bombardear con electrones cualquier punto de la pantalla.

Esta pantalla se encuentra absolutamente recubierta de una sustancia fosforescente, que tiene la propiedad de iluminarse cuando los electrones acelerados chocan contra ella.

Así, si deflectamos el haz de izquierda a derecha con suficiente velocidad y al llegar al extremo rápidamente lo llevamos de nuevo a la izquierda, veremos una recta horizontal en la pantalla.

Si además de hacer retroceder el haz, lo desplazamos verticalmente, cada vez que llega al borde y, al acabar repetimos el proceso indefinidamente, nos encontraremos con una pantalla totalmente blanca. El efecto recuerda a dos bucles FOR-NEXT anidados. (En el próximo número veremos un ejemplo concreto.)

En las televisiones europeas el «refresco» de la imagen se realiza 25 veces por segundo (cada 40 ms).

Es necesario repetir rápidamente los barridos por dos motivos: 1) Permitir una ilusión de movimiento. 2) Cuando el haz alcanza un punto y posteriormente lo abandona, dicho punto no permanece indefinidamente iluminado sino que, paulatinamente va perdiendo su luz (en un proceso típicamente exponencial).

El último experimento consistiría en variar la intensidad del haz. De esta manera podemos crear infinidad de tonos y contrastes. Así resulta posible representar textos o imágenes.

Un lápiz óptico puede detectar una zona oscura o iluminada.

Cuando nosotros vemos un punto iluminado, no es tal, en realidad se trata de un punto que se enciende y apaga 25 veces por segundo, lo suficientemente rápido como para parecer continuo.

Pero la electrónica es más rápida que nuestro ojo y dicho parpadeo puede ser fácilmente detectado. Y lo más interesante, nuestro lápiz no va a detectar un determinado nivel de luz que puede ser muy variable (recordemos que la pantalla va a estar siempre iluminada), sino las raudas variaciones de intensidad luminosa.

Ya tenemos pues el sistema. Debemos usar algo que detecte esos pulsos luminosos y que los amplifique. Queda claro que, si nos encontramos sobre una zona clara recibiremos una señal pulsante de 25 Hz mientras que, si apuntamos a una zona oscura captaremos la más absoluta nada. Ahora lo que debemos pensar es lo que vamos a hacer con la señal que nos ofrece el lápiz.

Hasta aquí, todos los sistemas en los que se basan los diferentes lápices ópticos coinciden. Las siguientes etapas marcan las diferencias fundamentales.

La opción más comúnmente usada destaca por su velocidad. Desgraciadamente, la implantación en el Spectrum ha de venir acompañada de grandes complicaciones. Los circuitos integrados especialmente concebidos para el tratamiento del video, acostumbran a incorporar una entrada para lápiz óptico que simplifica enormemente las cosas.

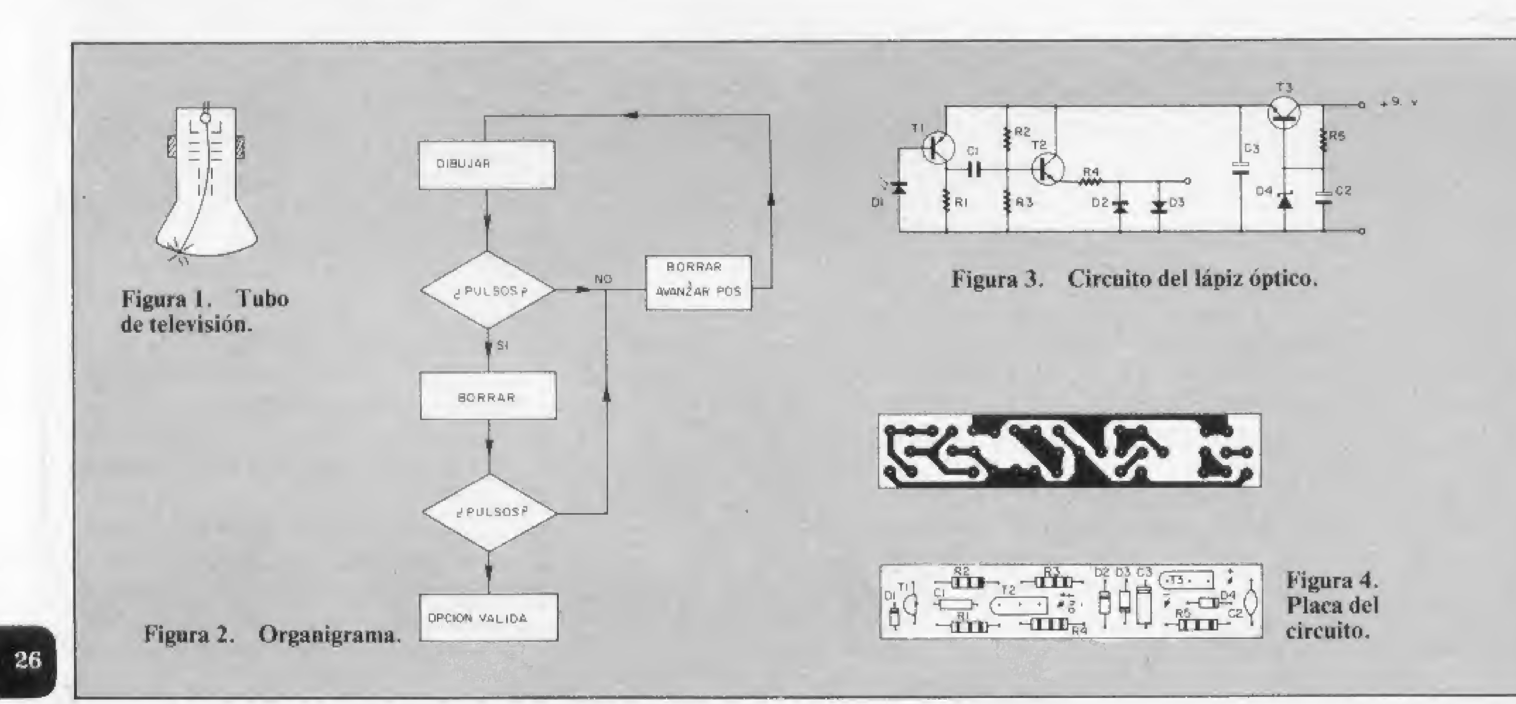


Figura 1. Tubo de televisión.

Figura 2. Organigrama.

Figura 3. Circuito del lápiz óptico.

Figura 4. Placa del circuito.

Para ser breves digamos que el punto hacia el que enfila el lápiz se determina en función de la celda de memoria que se está enviando al circuito de video para su representación en pantalla. En estos casos la pantalla debe tener un color claro.

El lápiz que nos ocupa no necesita enchufarse al conector de expansiones. Podemos usar una entrada ya existente: la clavija de EAR. Concebida inicialmente para la entrada de la información grabada en cinta, nada nos impide usarla para otros fines.

Lo único que habremos de hacer, será amplificar suficientemente la señal recibida por el fotodetector y enviarla al ordenador a través de la entrada de cinta magnética. Simple ¿no es cierto? Explicaremos brevemente ahora el método usado para hacer posible el uso del lapiz.

Supongamos que, trabajamos con papel negro y tinta blanca y queremos seleccionar la opción de un menú. Iremos dibujando cuadrados blancos al frente de cada opción hasta que se detecte alguna señal del lápiz. Pero, ¿será que el lápiz se halla sobre el punto dibujado, o sobre cualquier otro sitio iluminado? El mejor método para comprobarlo, es borrar el cuadrado testigo para ver si ahora dejamos de detectarlo. Si sigue dando señales, es que el lápiz se encuentra sobre una superficie luminosa de origen desconocido. Continuamos con el ciclo.

Si por contra no se recibe señal, podemos bien repetir la operación para estar totalmente seguros, bien dar la respuesta como válida. Sencillo ¿no? En un próximo número, mostraremos programas concretos que arrojarán más luz sobre el asunto.

Un punto iluminado sólo es un punto que se enciende y apaga 25 veces por segundo.

Vamos a meternos de lleno con el *hardware* necesario.

El circuito de nuestro lápiz óptico se muestra en la figura 3. El fotodiodo D1 es un traductor luz-corriente. El transistor T1 al que se halla directamente unido, actúa como amplificador de corriente.

En los extremos de la resistencia R1, tenemos una tensión proporcional a la luz que incide sobre la pantalla: iluminación natural (luz en 'continua'), iluminación artificial (en 'alterna') y la provocada por la pantalla de televisión iluminada (rápidos pulsos).

La resistencia R1 deberá determinarse experimentalmente en función del fotodiodo y transistor usados, así como de la luz ambiente normal. Sin embargo, para un BPW 22 y un BC 557 B como D1 y T1 respectivamente, una resistencia de 10K funcionó perfectamente en todos los prototipos.

El condensador C1 se encarga de realizar un filtrado conveniente, con objeto de eliminar perturbaciones procedentes de la iluminación artificial. También realiza el desacoplo necesario, eliminando la componente continua de la señal.

A continuación alrededor de T2 encontramos una etapa buffer (separadora) que tiene una impedancia de salida lo suficientemente baja como para que pueda conectarse a la entrada de *cassette*. Como quiera que las señales pueden alcanzar picos de hasta 5 voltios, se ha decidido poner a continuación una etapa limitadora. Además de mejorar el margen dinámico, impide que la señal alcance valores superiores a 0,6 V de pico.

El lápiz óptico recibirá su alimentación de la misma fuente que alimenta al Spectrum. Sin embargo, no podrá usarse directamente ya que es tremendamente 'sucia' y requiere un filtrado previo. El transistor T3 y componentes asociados (C2, C3, R5 y D4), constituyen una pequeña fuente de alimentación estabilizada a 9,4 V. Este circuito nos evita los voluminosos condensadores de desacoplo.

El conjunto puede montarse en una pequeña placa de 11 × 48 mm. (ver figura 4). No se ha previsto un interruptor ya que aumentaría excesivamente el volumen ocupado por el lápiz a no ser que se tratara de un modelo miniatura, necesariamente caro. Nada impide colo-

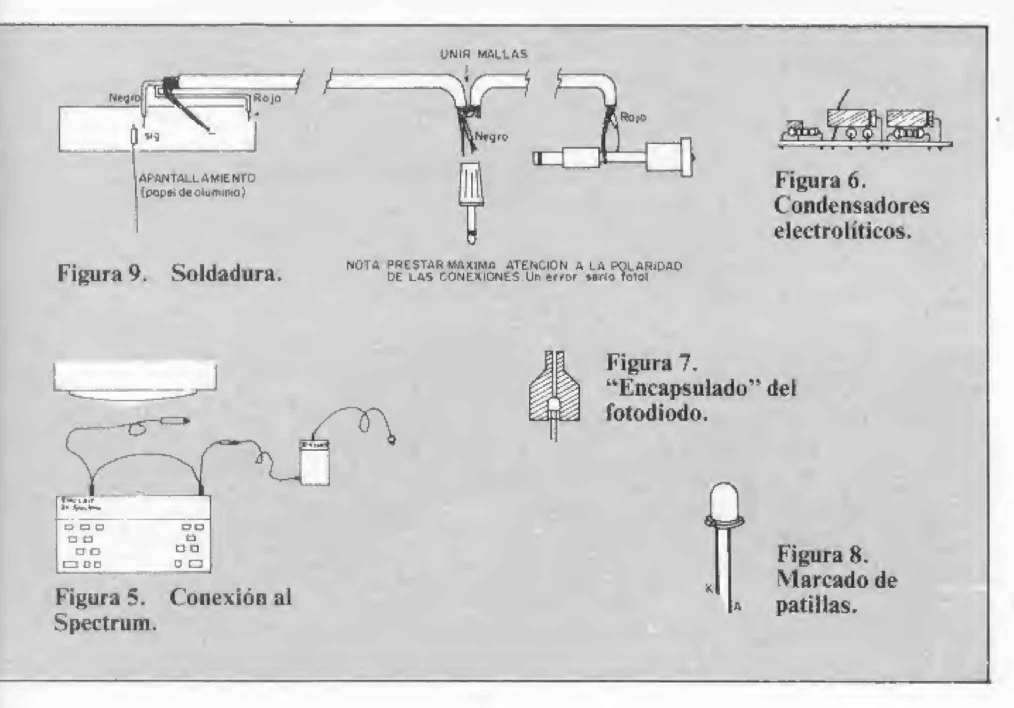


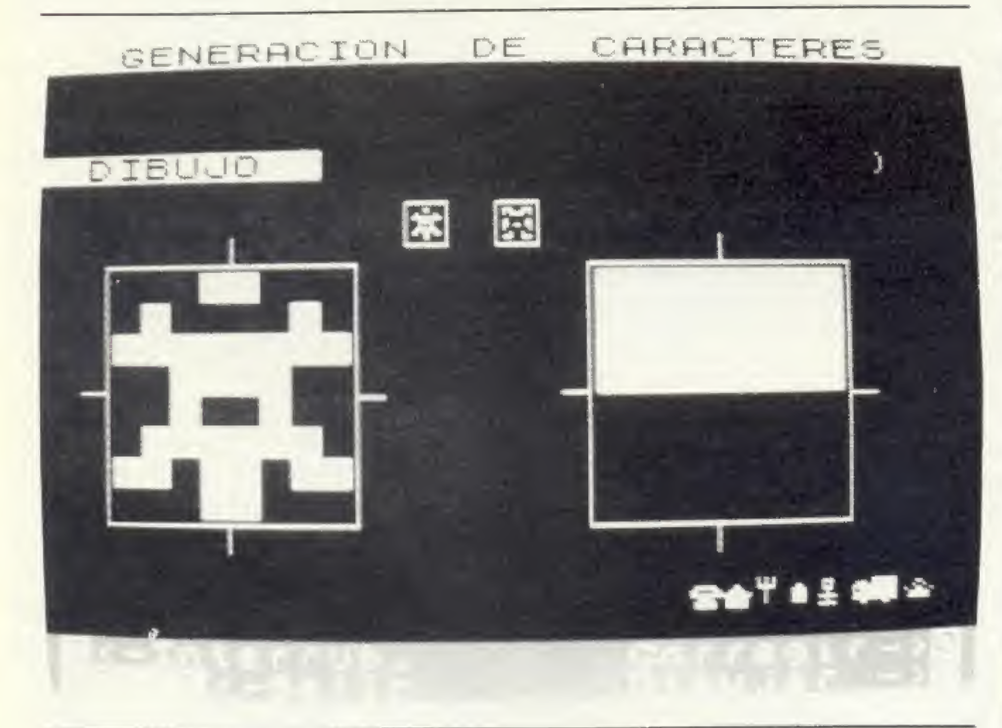
Figura 9. Soldadura.

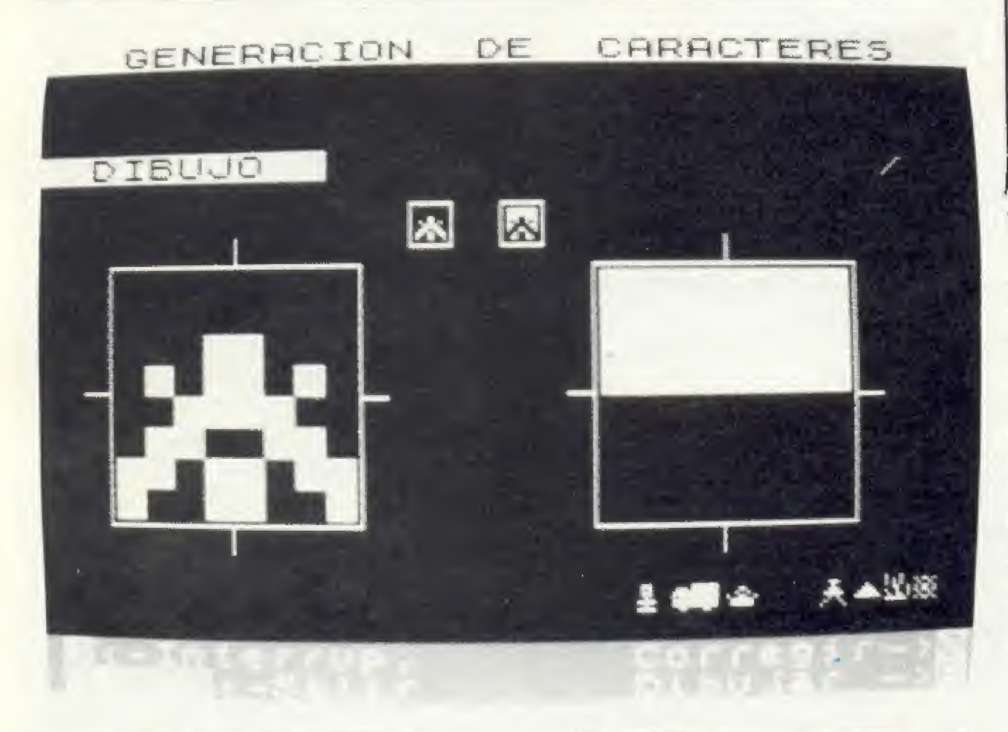
Figura 6. Condensadores electrolíticos.

Figura 7. "Encapsulado" del fotodiodo.

Figura 8. Marcado de patillas.

Figura 5. Conexión al Spectrum.





carlo en serie con la alimentación.

El conexionado del conjunto se muestra en la figura 5.

Montaje

En primer lugar, se soldarán las resistencias y el condensador C1. A continuación pasaremos con los semiconductores: diodos D2, 3 y 4 y los transistores. Prestar atención a la orientación adecuada. Por último, se montarán los condensadores electrolíticos que irán por encima de los componentes tal como muestra la figura 6.

El fotodiodo necesita un proceso de 'encapsulado'. Con objeto de hacer que, el fotodiodo resulte iluminado únicamente por aquellos puntos de la pantalla a los que está apuntando, se ha de fabricar una estructura como la de la figura 7. Podrá utilizarse cualquier material que cumpla la condición de ser opaco y aislante. El «agujero» deberá practicarse con limpieza y no deberá rallarse el encapsulado del diodo. Recordamos que, antes de introducir el diodo en su 'casa', deberemos proceder al marcado de sus patillas con objeto de soldarlo correctamente. (Tener en cuenta que toda marca en un diodo corresponde al cátodo o que la patilla más larga es el ánodo). Ver figura 8.

Los cables usados deberán ser apantallados de doble conductor: positivo de la alimentación, señal y malla a masa. El conjunto del lápiz deberá estar también apantallado con papel de aluminio y enrollado con éste, se deberá introducir un hilo desnudo conectado a masa. Por razones obvias, se aislará el circuito con un papel o plástico con objeto de evitar cortocircuitos.

Recibe la alimentación de la misma fuente que alimenta al Spectrum.

El lápiz se alojará en algún tubo de material plástico o metálico (resultan ideales los rotuladores tipo Edding 3000 ya gastados). Los cables y conectores se soldarán como indica la figura 9.

Lista de componentes:

R1: 10 K (ver texto).
R2: 68 K.
R3: 16 K.
R4: 560 Ω.
R5: 20 K.
T1: BC 558.
T2, T3: SC 149.
C1: 10 μF.
C2: 1 μF 15 V.
C3: 10 μF 15 V.
D1: porodiodo de si (BPW 22).
D2, D3: 1N4148.
D4: diodo tener 10 V, 1/4 W.

Luis Miguel

El SITI o Sistema Integrado de Tratamiento de Información es un programa que permite "integrar" la Base de Datos y la hoja de cálculo. Es decir, se puede guardar, seleccionar información y efectuar cálculos entre diferentes partes de un fichero. Lógicamente la información puede quedar grabada, bien en *cassette* o bien en *microdrive.* Para mayor simplicidad, el propio programa selecciona el *microdrive* como unidad de almacenamiento, siempre que esté conectado el *interface* 1.

El programa se ofrece en *cassette* con un completo manual de instrucciones, pero de difícil lectura. Completo porque a través de sus 20 hojas se ofrece amplia información para el manejo de las distintas opciones del programa con profusión de ejemplos, pero demasiado "oscuro", para evitar las copias. Se ha utilizado un papel de color marrón oscuro, de forma que el propio original —imprescindible para trabajar con el programa— resulta de difícil lectura.

No sólo en el manual hay ejemplos. Al cargar el programa, éste (recuerde que no es válido para el Spectrum de 16K) se ejecuta automáticamente y contiene un ejemplo con once fichas, en las que se combinan sus dos características principales: campos numéricos y alfanuméricos y campos que son el resultado de efectuar cálculos sobre otros. Siempre en el formato de 64 caracteres por línea. La figura 1 reproduce el menú de opciones. La estructura de esta primera pantalla se arrastrará en todos los demás, es decir, las tres líneas inferiores dan información sobre el programa en general o sobre la opción utilizada en cada momento: nombre del archivo, número de fichas existentes y seleccionadas, opción utilizada, forma de finalizar y memoria libre. Sobre estas líneas se desarrolla el programa.

Para grabar datos, primero hay que determinar su número y tipo, es decir, hay que realizar la ficha de entrada de datos. Esta es la primera ventaja de toda base de datos: su flexibilidad. Usted mismo se configura "su ficha" de acuerdo a sus necesidades. La figura 2 muestra la ficha que viene como ejemplo. Se comenzó por una referencia. Después precio de venta, precio proveedor, cantidad stock, stock mínimo, condiciones y descripción del artículo. Otros campos definidos y calculados a partir



Analizamos el SITI

Uno de los programas de gestión más necesarios para cualquier actividad lo constituye la Base de Datos. Un buen programa de base de datos ha de poder guardar todo tipo de información y, lo que es más importante, poder recuperarla bajo distintos criterios de búsqueda y localización. La interrelación de la información y la rapidez de acceso a la misma según criterios de búsqueda, es un punto clave de toda Base de Datos. La memoria del ordenador juega aquí un papel primordial. Teniendo en cuenta las limitaciones de memoria del Spectrum, el programa de Ventamatic —12 Kbytes en código máquina— que le presentamos en esta ocasión, aprovecha al máximo sus posibilidades.

de los anteriores, fueron bajo mínimo, importe total a PVP, total acumulado, etc. A continuación de cada uno de estos títulos se especifica el tipo de campo, es decir, el número máximo de dígitos a introducir y si estos serán numéricos o alfanuméricos. Los numéricos se determinan por el símbolo ‡, los alfanuméricos por ▭ y las fechas por el símbolo de la libra £. Así, siguiendo con el ejemplo de la figura 3, la referencia, condiciones y descripción del artículo se definen como campos alfanuméricos y como numéricos los restantes. En el caso de los datos numéricos, el "punto" sirve para separar los dígitos, quedando alineados por la derecha. Dado que estamos ante un programa realizado en nuestro país, se sigue la nomenclatura de separación española del "punto" para miles y la "coma" para los decimales, al contrario de la mayoría de programas anglosajones. La única precaución a tener presente es no dejar más de un espacio de separación entre los titulares de los campos, ya que es este precisamente la identificación que utiliza el programa como separador.

Cada vez que complete la introducción de datos de una ficha, presionando EDIT quedará almacenada. En nuestro ejemplo, se almacenan los datos introducidos en la parte superior de la pantalla, junto con los que el programa "calcula" en la parte inferior de la pantalla (figura 2). Para ver esto con mayor claridad, compare la figura 2 con la 3. La única diferencia está en los campos que aparecen "encerrados" entre barras verticales. De esta forma, el programa identifica los campos cuyo contenido ha de calcular en función de otros campos, es decir, de esta forma el programa se convierte en una nueva versión de las clásicas hojas de cálculo, ya que lógicamente los cambios en las casillas o campos superiores siempre producirán un cambio en los restantes campos referenciados a dichas casillas. Siguiendo el ejemplo (figura 2), a la derecha del titular *Bajo mínimo* aparecerá un 1 si el resultado de restar el valor contenido en el campo *Cantidad stock* del campo *Stock mínimo* es menor que cero. Para entender esto, hay que tener

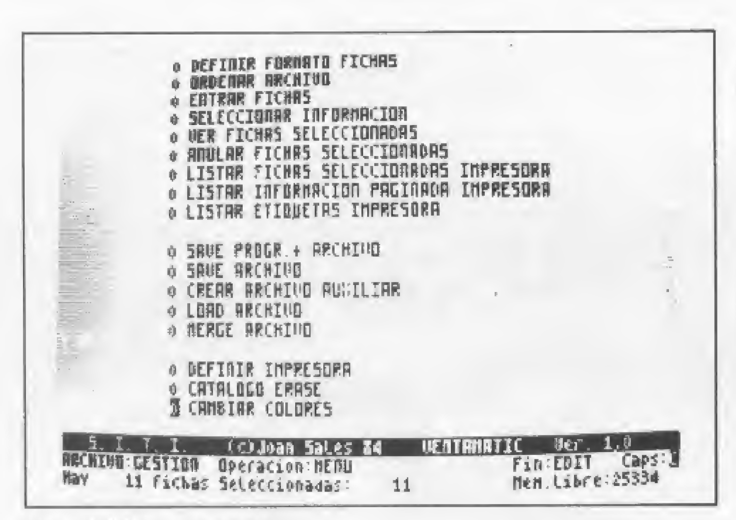


Figura 1. Menú de opciones.

```
CATALOGO DE PROGRAMAS DE GESTION PARA ZX-SPECTRUM
Ref.:            Precio venta:     ‡.‡‡‡.‡‡‡ ptas
                 Precio proveedor: ‡.‡‡‡.‡‡‡ ptas

                 Cantidad stock: ‡‡.‡‡‡ unidades
                 Stock minimo:   ‡‡.‡‡‡ unidades

                                      Bajo minimo:‡

|Bajo minimo=((Cantidad-Stock)<0)|
Condiciones:     DESCRIPCION:
|Importe total a P.V.P.=Precio venta×Cantidad|
Importe total a P.V.P.:   ‡.‡‡‡.‡‡‡ ptas
|Total ac=Total ac+Importe|
Total acumulado a P.V.P:  ‡.‡‡‡.‡‡‡ ptas
|Importe total a P.prov=Cantidad×Precio proveedor|
Importe total a P.prov.:  ‡.‡‡‡.‡‡‡ ptas
|Total a P.prov=Total a P.prov+Importe total a P.prov|
Total a P.prov.:          ‡.‡‡‡.‡‡‡ ptas
S. I. T. I.   (c)Joan Sales 84   VENTAMATIC   Ver. 1.0
ARCHIVO:GESTION Operacion:DEFINIR FORMATO  Fin:EDIT  Caps:
May   11 fichas Seleccionadas:  11         Mem.Libre:25334
```

Figura 2. Ejemplo de una ficha.

algún conocimiento del Spectrum y del programa:

1) El Spectrum, puede trabajar con expresiones lógicas del tipo LET B=(A > 0), lo que quiere decir que a B le asigna el valor 1 si el contenido del paréntesis es cierto o el valor 0 en caso contrario. Es decir, B=1 si A > 0 y B=0 si a ‡ 0.

2) Para la identificación de los campos, el programa no necesita que se haga referencia al nombre completo. En el ejemplo, para referirnos al *Precio venta* sería necesario decir al menos *Precio v* para diferenciarlo del *Precio proveedor.*

Siguiendo el mismo razonamiento el *Importe total a P.V.P.*, se obtiene de multiplicar el *Precio venta* por la *Cantidad.* Análogamente se obtienen los restantes campos (figura 3).

Una vez definida la ficha e introducidas éstas. Ya dispone de una Base de Datos. Decíamos que la mejor característica es su flexibilidad, que puede verse en sentido doble, flexibilidad para configurar las fichas acorde a las necesidades individuales, y flexibilidad para obtener información según distintos criterios de clasificación. A esta segunda posibilidad se refiere la opción "SELECCIONAR INFORMACION" del menú, aunque el manual lo describe como "SELECCIONAR FICHAS". En esta opción, hablar de flexibilidad es quedarse cortos. Se puede trabajar con diez criterios distintos de clasificación:

Selección en campos alfanuméricos

La selección no tiene en cuenta los espacios a la derecha del campo, siendo indiferentes minúsculas

Qué es una Base de Datos

Una Base de Datos es un programa de gestión de la información que sirve para almacenar y consultar información, manipularla y clasificarla en un diverso grado, así como utilizarla para generar informes o listados, entre sus principales funciones.

Hasta hace solo unos pocos años, estos programas eran tan complejos que sólo los podíamos utilizar profesionales informáticos, o bien tan simples que el control que se tenía sobre la información que se deseaba organizar y manipular era muy limitado.

Hoy la situación ha cambiado considerablemente y pequeños ordenadores como el Spectrum disponen ya de programas de gestión de la información que merecen el calificativo de Base de Datos. Entre estos, limitados en gran medida por las propias características del ordenador, y los que corren en un *mainframe,* existe una amplia diversidad de programas que se pueden clasificar en tres: manejo de ficheros, programas de nivel intermedio y sistemas de Bases de Datos avanzados. A estos hay que añadir los paquetes integrados.

Los programas de manejo de ficheros llevan a cabo las funciones básicas de almacenamiento de la información, búsqueda, manipulación y algunos cálculos sencillos, y son capaces también de generar informes en base a los datos de un fichero. Dado que conceden importancia a la sencillez de manejo sobre lo demás, sus posibilidades suelen estar gravemente limitadas en flexibilidad, en grado de adaptación a las necesidades concretas del usuario.

Los programas de nivel intermedio resultan adecuados para aplicaciones como la contabilidad general, sistemas de control de ventas, control de almacen, etc. Muchos de ellos pueden acceder a más de un fichero a la vez y clasificar registros en base a claves múltiples. A diferencia de los anteriores, almacenan, ordenan, manipulan, localizan datos, y preparan informes con una mayor flexibilidad.

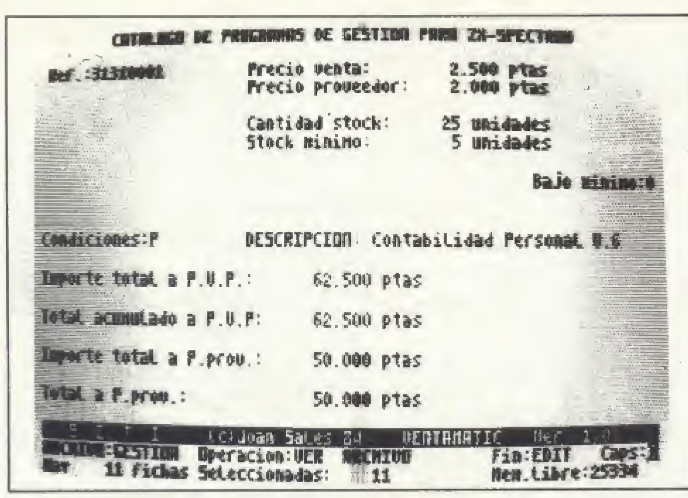


Figura 3. Ejemplo de formato de una ficha.

La figura 1 muestra las opciones del programa, convenientemente separadas en tres bloques. Las figuras 2 y 3 corresponden al ejemplo que viene con la cinta. El programa se autoejecuta, mostrando el menú de opciones, en cuya línea de estatus informa del número de fichas existentes y las seleccionadas.

o mayúsculas. Los criterios de selección son:

1. Contenido exacto del campo. Supongamos que se escribe "pilas" en el campo *Descripción.* De esta forma se seleccionarían todas las fichas que contuviesen solamente la palabra "pilas" en dicho campo.
2. Contenido de primeros caracteres. Se selecciona un campo solo por los primeros caracteres antes del punto. Por ejemplo, si en la *referencia* se colocase "DJ." se seleccionarían todas las fichas que en dicho campo comenzasen por "DJ".
3. Contenido en cualquier posición. Si el "punto" se coloca al principio, ello denota la búsqueda de las fichas que contengan la información que se especifica después, ya sea al principio, en medio o al final del campo.
4. Cualquier caracter en una posición. A diferencia del anterior, se puede dejar variable un caracter dentro de una clave de búsqueda, utilizándose para ello el símbolo "?". Así la clave "p?so" selecciona fichas que contengan en el campo elegido la palabra piso, paso, peso, poso, etc.
5. Campos vacíos. Incluso si se dejan campos vacíos, ello puede servir por sí solo para localizar las fichas, al introducir el signo de subrayado "–" como criterio de búsqueda en un campo.
6. Significado contrario. En cualquiera de los casos anteriores, sirve para buscar "lo contrario" de los especificado. Así "/P." buscaría todas las fichas en las que el campo especificado NO comienza por "P".
7. Selección por orden alfabético. Independientemente de que las fichas se encuentren ordenadas, se puede clasificar alfabéticamente. "<D" selecciona las fichas cuyo campo especificado comience por la inicial A, B ó C.

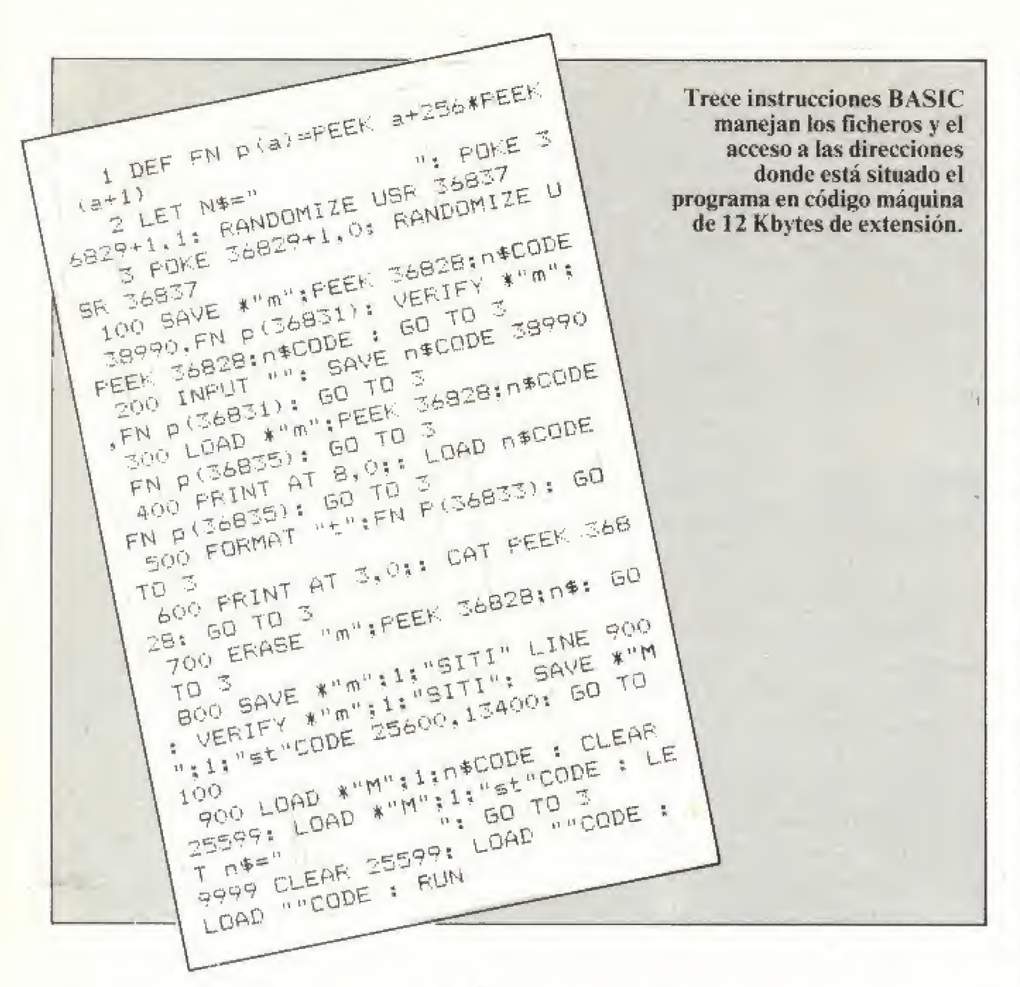
Selección en campos numéricos

Siguiendo con el ejemplo que acompaña al programa (figuras 2 y 3), la selección de campos numéricos puede realizarse de la siguiente forma:



Los sistemas de Bases de Datos avanzados disponen de potentes comandos con los que crear aplicaciones particularizadas. Están concebidos para procesar grandes cantidades de datos, y utilizar de manera coordinada varios ficheros independientes, con lo que se consigue presentar aspectos diferentes de la misma información.

Finalmente, los paquetes de *software* integrado se componen de diversos programas, generalmente gráficos, hoja de cálculo, procesador de textos y Base de Datos. El ejemplo típico consiste en una hoja de cálculo que incorpora algunas capacidades de gestión de Base de Datos como la ordenación o selección de información.



```
   1 DEF FN p(a)=PEEK a+256*PEEK
(a+1)
   2 LET N$="                ": POKE 3
6829+1,1: RANDOMIZE USR 36837
   3 POKE 36829+1,0: RANDOMIZE U
SR 36837
 100 SAVE *"m";PEEK 36828;n$CODE
38990,FN p(36831): VERIFY *"m";
PEEK 36828;n$CODE : GO TO 3
 200 INPUT "": SAVE n$CODE 38990
,FN p(36831): GO TO 3
 300 LOAD *"m";PEEK 36928;n$CODE
FN p(36835): GO TO 3
 400 PRINT AT 8,0;: LOAD n$CODE
FN p(36835): GO TO 3
 500 FORMAT "t";FN P(36833): GO
TO 3
 600 PRINT AT 3,0;: CAT PEEK 368
28: GO TO 3
 700 ERASE "m";PEEK 36828;n$: GO
TO 3
 800 SAVE *"m";1;"SITI" LINE 900
: VERIFY *"m";1;"SITI": SAVE *"M
";1;"st"CODE 25600,13400: GO TO
100
 900 LOAD *"M";1;n$CODE : CLEAR
25599: LOAD *"M";1;"st"CODE : LE
T n$="                ": GO TO 3
9999 CLEAR 25599: LOAD ""CODE :
LOAD ""CODE : RUN
```

Trece instrucciones BASIC manejan los ficheros y el acceso a las direcciones donde está situado el programa en código máquina de 12 Kbytes de extensión.

8. Contenido igual que. Se podría seleccionar todos los artículos que den *Bajo mínimo* o dicho de otra forma, que den el valor 1 en el campo *Bajo mínimo.* "¡Su listado podría enviarse directamente al proveedor!".
9. Contenido mayor que y menor que. Imagínese un nuevo campo en la ficha que fuese el margen de beneficios por producto. Fácilmente podría sacarse la clasificación entre aquellos productos que diesen un margen "mayor" o "menor" a una determinada cantidad.
10. Significado contrario. Idéntico a los campos alfanuméricos.

La selección sirve, como su propio nombre indica, para seleccionar un número de fichas que cumplan con los requisitos de la clave de selección. Ello no afecta a las fichas. Lo que si las puede modificar es la opción "ORDENAR ARCHIVO". Eligiendo esta opción, volverá a aparecer el formato de la ficha para seleccionar el criterio de selección. Al ordenar por un campo determinado, se respeta el orden anterior que pudiera haber en otros campos. Así, si se ordena un archivo por *Descripción* y luego por *Precio,* el orden final será por *Precio* de menor a mayor y dentro de los artículos con el mismo precio por orden albafético de *Descripción.* Para la selección del campo-clave para la ordenación, simplemente se coloca un caracter en la primera posición del campo.

Es interesante destacar la posibilidad de trabajar con fechas, no contemplado en el ejemplo que viene con el programa. Un campo con el formato £ debe tener siempre seis dígitos para la introducción de la fecha según el formato DDMMAA. De esta forma la ordenación del archivo por fechas se realiza correctamente.

Por supuesto, se incluyen también las opciones de actualización no sólo de entrada de nuevas fichas, sino de anulación y modificación. Estas opciones vuelven igualmente al formato definido de ficha, sobre el que se realizan las actualizaciones con facilidad y rapidez.

Finalmente, la impresión. No es necesario contar con impresora, al igual que tampoco son necesarios los *microdrives.* Las fichas seleccionadas pueden visualizarse en pantallas con la ayuda de potentes comandos ("Enter" para paso a la siguiente ficha, "8" para avanzar cinco fichas, "A" anula la ficha del archivo, "U" pasa a la última ficha, "M" para modificaciones...). Claro que siempre puede ser conveniente tener una copia impresora o un *hard copy* por seguir la terminología al uso. La opción del menú "Definir impresora" tiene tres posibilidades: (1) ZX *printer* o GP-50 de Sheikosa, (2) *Interface* Centronics y (3) *Interface* 1 para impresoras serie, mediante el *interface* 1 (como ya indicábamos al inicio de este artículo, este mensaje sólo aparecera si está conectado el *interface* 1).

Veamos en detalle cómo ha de conectarse a dos de los *interface* Centronic más comunes: Si conecta el *interface* del tipo Centronics de **Ventamatic,** no cargue el programa que incorpore dicho *interface:* el SITI no funcionaría. Si utiliza el de **Indescomp,** siga los siguientes pasos:

1. Cargue el programa. LOAD * "M"; 1; "SITI".
2. Seleccione la opción del menú MERGE ARCHIVO.
3. Pulse ENTER dos veces y BREAK. El programa se detendrá con el mensaje BREAK - CONT repeats 400:2.
4. Ejecute POKE 34934,4
5. Pulse RUN. Generalmente suele funcionar.

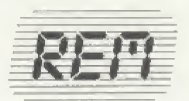
Para finalizar, el manual informa del futuro "modulo adaptador" para el context V.6, (procesador de textos desarrollado por la misma casa de software), que permitirá hacer cartas personalizadas con la información que proporcione un archivo grabado con el SITI, y otro módulo de conversión de archivos del Masterfile (célebre base de datos inglesa que comentaremos en un próximo número) al SITI.

Guía del comprador de Todospectrum



ESPECIALISTAS EN SINCLAIR
IMPRESORAS. MONITORES.
PERIFERICOS
PROGRAMAS EDUCATIVOS.
GESTION. OCIO...
AMPLIACIONES DE MEMORIA

C/Silva, 5 - 4.° - Telf. 242 24 71
28013 MADRID



- Ordenadores personales Hard y Soft.
- Cursos de Basic.

Oficinas: **RENOVACION EN MARCHA, S.A.**
c/. Espronceda, 34 - 2º int. - MADRID-3
Teléfono **(91) 441 24 78**
Tienda: **REM SHOP 1**
c/. Galileo, 4 - MADRID-15
Teléfono **(91) 445 28 08**

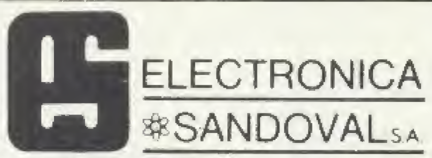


MULTISYSTEM, S. A.

BOUTIQUE INFORMATICA

* Ordenadores Personales.
* Micro-ordenadores de gestión.

Todas las novedades en:
Programas. - Periféricos - libros
(nacionales y de importación)
Para: Spectrum - Dragón - Base 64
Spectravideo - Oric - Commodore, etc.
C/ San Vicente, 53. **ALICANTE.** Tel. (965) 21 55 66.



DISTRIBUIDORES DE:
COMMODORE-64
ORIC-ATMOS
ZX SPECTRUM
SINCLAIR ZX 81
ROCKWELL'-AIM-65
DRAGON-32
NEW BRAIN
DRAGON-64
CASIO FP-200

ELECTRONICA SANDOVAL, S. A.
C/ SANDOVAL, 3, 4, 6. 28010-MADRID
Teléfonos: 445 75 58 - 445 76 00 - 445 18 70
447 42 01
C/ SANDOVAL, 4 y 6
Centralita 445 18 33 (8 líneas)



CASSETTES
DE CALIDAD PROBADA
PARA ORDENADORES

Cada uno	Caja de 10	Caja de 30
C-5 199 ptas	1 393 ptas	3 582 ptas
C-10 209 ptas	1 463 ptas	3 762 ptas
C-15 219 ptas	1 533 ptas	3 942 ptas
C-20 229 ptas	1 602 ptas	4 122 ptas

Libre de gastos de envío contra reembolso correos
CAMAFEO INC. Dep. 03
José Lázaro Galdiano, 1. 28036 Madrid.



DEBIDO A LA GRAN EXPANSION DE LA EMPRESA
CAMBIAMOS EL DOMICILIO SOCIAL A UNA NUEVA NAVE
DE 1.000 m. CON 3 PLANTAS

EN LA DIRECCION: Avda. MEDITERRANEO. 9
Tels. 433 44 58 (5 líneas) 433 45 48 - 433 48 76-28007 MADRID

SUSCRIBASE A Todospectrum (12 NUMEROS)

TARIFA DE PRECIOS DE SUSCRIPCION

	CORREO ORDINARIO		CORREO CERTIFICADO		CORREO AEREO		CORREO AEREO-CERTIF	
	PTAS.	$	PTAS.	$	PTAS.	$	PTAS.	$
ESPAÑA	3.000	21	3.273	23	3.055	22	3.333	24
EUROPA, MARRUECOS, TUNEZ, TURQUIA, ARGELIA Y CHIPRE	3.456	25	4.272	31	3.600	26	4.418	31
COSTA RICA, CUBA, CHILE, PARAGUAY Y REP. DOMINICANA	3.396	24	4.212	30	4.164	30	4.980	36
GIBRALTAR Y PORTUGAL	3.264	23	4.080	29	3.149	22	3.965	28
FILIPINAS	3.264	23	3.540	25	3.775	27	4.050	29
RESTO DEL MUNDO	3.456	25	4.272	31	4.224	30	5.040	36

CUPON DE PEDIDO

Recorte y envie este cupón a: **Todospectrum** EDISA. **Lopez de Hoyos, 141 - 28002 - MADRID**

El importe lo abonaré: POR CHEQUE □ CONTRA REEMBOLSO □
CON TARJETA DE CREDITO □ American Express □ Visa □ Interbank □

Número de mi Tarjeta: ________________ Fecha de caducidad: ________

NOMBRE ________________

DIRECCION ________________

CIUDAD ________________ D.P. ____ PROVINCIA ________________

EL LOGO EN LA ESCUELA

«Programar una computadora no significa ni más ni menos que comunicarse con ella en un lenguaje que tanto la máquina como el usuario humano puedan 'comprender'. Y aprender lenguajes es una de las cosas que mejor hacen los niños. Todo niño normal aprende a hablar. ¿Por qué no aprendería entonces a 'hablar' con una computadora?»
Seymour Papert. (Desafío a la mente.)

La premisa de diseñar un lenguaje que permitiera al niño programar al ordenador y no por el contrario, ser «programado» por éste (una de las consecuencias inevitables de la tan nefasta aplicación de la Enseñanza Asistida por Ordenador), dirigió la labor de Seymour Papert, director del grupo Logo del laboratorio de Inteligencia Artificial del Instituto Tecnológico de Massachusetts. Este matemático, que fue discípulo de Piaget en el Centro Internacional de Epistemología Genética de Ginebra entre los años 1959 y 1964, dedicó su interés al estudio del proceso de aprendizaje en los niños y la psicología de la inteligencia.

Para Seymour Papert, el aprendizaje más que un fenómeno de asimilación acumulativa, debe ser considerado como un mecanismo creativo en el cual el niño construye sus estructuras intelectuales a partir de los elementos que le suministra el entorno. Lo que el niño aprende son estrategias de resolución de problemas y modelos de interpretación del mundo. Las respuestas que producirá serán en relación a los problemas que le toque resolver, y sus modelos de interpretación estarán relacionados con aquellos elementos que ciertamente están a su alrededor. Las dificultades del niño en el aprendizaje de la matemática se deberían más bien a la falta de una vivencia empírica y de percepciones «sensoriales», que a dificultades conceptuales.

Adquiere así una especial importancia la percepción. Por lo que la creación de contextos en los cuales se pueda desarrollar una praxis, que dé como resultados la resolución de problemas y la formulación de interpretaciones, se convierte en una labor fundamental. En este tipo de situaciones la espontánea potencialidad de aprendizaje del niño se ve favorecida al acercarle a su mundo «familiar» conceptos abstractos y complejos; y al permitirle construir desde su propia perspectiva el entendimiento de los mismos.

Otro de los elementos de sustancial importancia en el aprendizaje es el propio desarrollo de la afectividad. Es innegable que cuando el niño se ve afectivamente ligado a los elementos que le motivan, los resultados de su aprendizaje mejoran. Una abierta predisposición a la realización de tareas que sean de su agrado y que estén subordinadas a los diferentes grados del desarrollo de su afectividad, proporciona un contexto que por estar hecho a su medida le resulta satisfactorio.



Esto permite que el estudio pase a ser más que una obligación un juego.

A partir de estas consideraciones se desarrolló el Logo como aquel instrumento que permitiría a los niños el uso del ordenador con la siguiente perspectiva: la introducción de su entorno de un elemento desencadenante de experiencias y creatividad y portador de un lenguaje lo más parecido al lenguaje natural que le permitiese dirigirlo y entenderlo.

En el lenguaje Logo el niño no manipula entidades lógicas abstractas. Aprende a dibujar conduciendo a una «tortuga», un pequeño animal que «vive» en la pantalla. Algunas tortugas Logo son robots que se mueven en el suelo mediante ruedas y que se encuentran unidas a la computadora por un cable o por rayos infrarrojos. La representación de la tortuga en la pantalla es un triángulo que ellos pueden dirigir con instrucciones tan naturales como avanza, retrocede, gira, etc.

Estos movimientos de la tortuga están vinculados al conocimiento previo que el niño posee sobre su entorno espacial. Por lo que la estrategia para realizar un proyecto del dibujo de una flor, por ejemplo, se convierte en el planteamiento de un algoritmo y en su resolución.

La relación del niño con la tortuga reúne así las siguientes ventajas:

— Es una relación lúdica. Y esto satisface exigencias de su desarrollo afectivo.

— El niño ejerce el control sobre el mundo de la tortuga. Lo hace crecer y desarrollarse en la medida en la que él mismo crece y desarrolla su pensamiento.

— El «error» en el aprendizaje adquiere un contenido diferente ya

que al dirigir la tortuga, las diferencias entre lo que le ordena y lo que sucede realmente se convierten en una fuente de comprensión más que de frustración.

Pero el Logo no debe entenderse exclusivamente como un lenguaje de programación para niños ya que la geometría de la tortuga va unida a una amplia serie de potencialidades: la posibilidad de definición de nuevos procedimientos (palabras) por el usuario que amplíen el microcosmos del Logo; la recursividad, que permite definir objetos en función de sí mismos; la manipulación sofisticada de palabras y listas; y por supuesto, la disposición de posibilidades de impresión, edición y comunicación, así como el almacenamiento de los procedimientos de manera sencilla.

El Logo es también un buen instrumento para aquellos padres que quieren que sus hijos se inicien en el uso del microordenador de una forma más positiva que la simple actitud de pulsar botones para aniquilar una flotilla de extraterrestres o para hacer que el personaje recorra decenas de pantallas recolectando objetos y sorteando obstáculos.

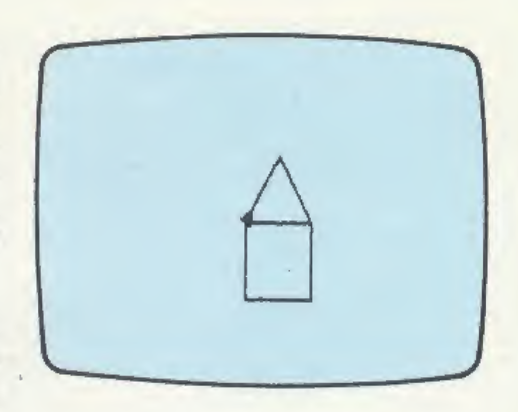


Figura 1. Comandos para comunicaciones.

COMANDOS PARA COMUNICACIONES

Nombre de la Instrucción	Descripción
.SETSERIAL n	Toma el valor de n en baudios y establece la velocidad de transmisión. Los valores de n pueden ser: 50, 110, 300, 600, 1200, 2400, 4800, 9600 o 19200.
.SERIALIN	Lee todo lo que llega al **port** RS232 en el valor en baudios establecido y nos da un **byte** entre 0 y 255.
.SERIALOUT n	Envía un **byte** (n) al **port** RS232.

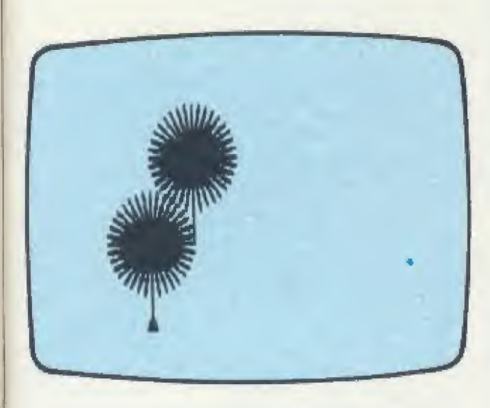


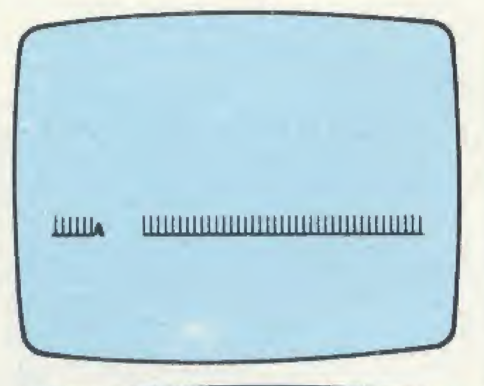
Otra de las importantes pretensiones de Seymour Papert en el diseño del Logo era que éste pudiera funcionar en un microordenador con una gran flexibilidad, con una buena potencia de gráficos y cuyo precio fuera asequible a las escuelas y a las familias.

La aparición del Logo de Sinclair para el ZX Spectrum de 48 Kbytes, a tan sólo tres años de la primera versión realizada para un microordenador, el TI99/4 de la casa **Texas Instruments Inc.,** ayuda a satisfacer este deseo y posibilita que los cientos de miles de usuarios del Spectrum en Europa puedan acercarse a este fascinante lenguaje.

El logo de Sinclair

Realizado por **Les Systems d'Ordinateurs Logo International** (París) y **Logo Computer Systems Inc.** (Quebec) para el Spectrum de 48 Kbytes, la versión inglesa, que es la que comentaremos, se presenta en una **cassette** y viene acompañada por dos manuales: El Logo de Sinclair 1 (gráficos de la tortuga) y el Logo de Sinclair 2 (Manual de Referencia a la Programación). Sabemos que una adaptación al castellano ya está en marcha.





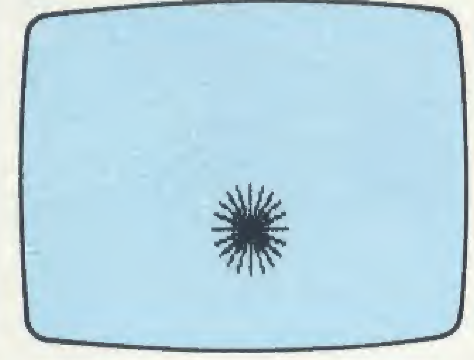
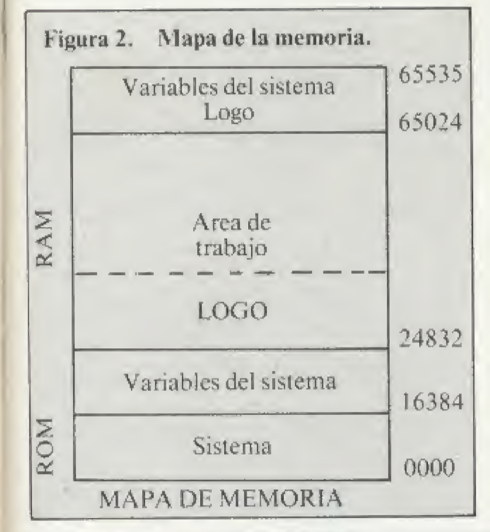


Figura 2. Mapa de la memoria.



En la elaboración de los manuales se descubre también la finalidad didáctica del Logo. A diferencia de muchos otros manuales incomprensibles para los usuarios, aquí se distingue con claridad entre los iniciados, a los que va destinado el primer volumen, y los que ya tienen una cierta experiencia de programación en Logo para los cuales la lectura del segundo nivel será suficiente.

El Logo de Sinclair 1 Gráficos de la tortuga no es una guía completa para el usuario en donde se exponga todo el vocabulario, sino que se concentra en la programación de la tortuga para dibujar, y permite la edición y el almacenamiento del trabajo realizado. La introducción del vocabulario Logo, además de ser gradual, está extensamente documentada con ejemplos y con la propuesta de proyectos para llevar a cabo. Los gráficos de la tortuga nos permiten ver claramente «lo que estamos haciendo mientras lo estamos haciendo», por lo que se convierte en una excelente introducción a la programación.

El Logo de Sinclair 2 nos da una concisa descripción de todos los términos primitivos del Logo, nos ofrece un resumen de la gramática del Logo y abunda en todo lo que no fue tratado en el primer volumen. Todo esto con ejemplos que permiten ilustrarnos suficientemente.

El Logo de Sinclair no es sólo un lenguaje, constituye un sistema operativo completo en la medida del microordenador para el cual está concebido. Nos permite utilizar todos los periféricos disponibles para el Spectrum: *cassette, microdrive* e impresora; podemos usar plenamente el *interface* RS232; e incluso reservar un espacio de memoria en el que implementar «ampliaciones» al Logo.

El programa (que ocupa 24 Kbytes) una vez cargado deja un «área de trabajo» de aproximadamente 7 Kbytes en la que se alma-

Figura 3. Comandos para el almacenamiento y carga de datos.

PROGRAMA PARA EL ALMACENAMIENTO Y CARGA DE DATOS

Nombre de la instrucción	Descripción	Significado
SAVEALL " Nombre del fichero.	Guarda todo lo que hay en el área de Trabajo.	GUARDATODO
SAVED " Nombre del fichero.	Guarda todo lo que hay en el Editor.	GUARDA EDITOR
SAVESCR " Nombre del fichero.	Guarda todo lo que hay en pantalla.	GUARDA PANTALLA
LOADD " Nombre del fichero.	Carga todo lo guardado por SAVED.	CARGA EDITOR
LOADSCR Nombre del fichero.	Carga todo lo guardado por SAVESCR	CARGA PANTALLA
CATALOG (sólo para *microdrive)*	Nos da el nombre del cartucho y el contenido (equivale a CAT)	CATALOGO (del *microdrive*)
ERASE "Nombre de fichero, Tipo de fichero.	Borra el fichero nombrado del *microdrive*. Los tipos son LOG, BIN (binario), SCR (pantalla), TXT (editor).	BORRA (en el *microdrive*)

cenarán los procedimientos y variables que vayamos creando o aquellos, que ya definidos, se carguen desde el *cassette* o el *microdrive*. Ver figura 2.

Uno de los aspectos más poderosos del Logo es su capacidad para trabajar con «procedimientos». Hay dos clases de procedimientos: aquellos a los que el Logo «conoce» y están presentes cada vez que cargamos el Logo de Sinclair, a los que llamamos primitivos, y aquellos que nosotros podemos definir.

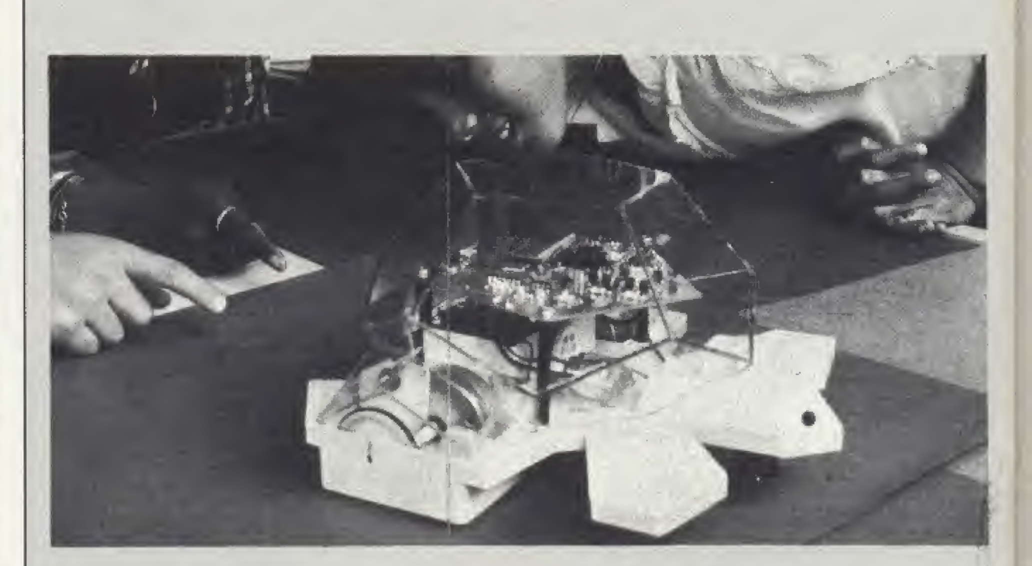
La zona de almacenamiento de los procedimientos definidos por el usuario no se reduce a estos 7 Kbytes de memoria. Existen comandos que permiten guardar y recuperar del *cassette o microdrive*, tanto procedimientos discretos como todos los que contenga el área de trabajo. Los comandos para guardar y leer del *microdrive* son mucho más sencillos en el Logo que en el BASIC de Sinclair. Basta asignar el canal con SETDRIVE del 1 al 8 para el *microdrive*. El canal 0 será para el *cassette*. La sintaxis es siempre LOAD, SAVE y sus variantes. Ver figura 3.

La introducción RESERVE n, reserva una zona de n *bytes* a partir de la dirección 65024 (hacia abajo), en la cual podemos cargar rutinas en código máquina. La carga de estas rutinas se gestiona mediante el comando .BLOAD, con los parámetros de dirección de carga y longitud. Análogamente existe .BSAVE. La instrucción .CALL dirección, ejecuta un programa en código máquina cargado a la dirección referida.

El uso de la impresora en el Logo de Sinclair es también posible. COPYSCREEN copia todo lo que se encuentra en la pantalla ya sean gráficos o textos. PRINTON activa la impresión de todo lo que escribimos en pantalla mientras que PRINTOFF anula esta facultad.

El port RS232 se gestiona con una serie de instrucciones que permiten modificar la velocidad de transmisión y enviar o recibir caracteres. Ver figura 1.

Miguel Figini,
Manuel Tristán
y Paco Riviere



COMO DIBUJA LA TORTUGA

El aprendizaje de los comandos de la tortuga es muy sencillo y el resultado de su uso se observa inmediatamente. Una vez cargado el Logo, tecleando MUESTRATORTUGA, ésta aparece en el centro de la pantalla orientada hacia arriba. La instrucción AVANZA 10 (o abreviado AV 10) provocará un desplazamiento de 10 pixels hacia arriba. La tortuga dibuja una línea recta en su recorrido. Del mismo modo podemos ordenarle RETROCEDE 10 (RT 10) y volverá al punto de partida sin borrar el rastro dejado. Con GIRADERECHA 90 (GD 90) veremos que la tortuga gira 90° a su derecha. Análogamente existe GIRAIZQUIERDA (GI).

Con estas cuatro instrucciones básicas podemos orientar la tortuga en cualquier dirección y hacerla avanzar o retroceder la distancia deseada sin más que indicarle el ángulo o la distancia a recorrer que queramos. Podemos hacer que no deje «rastro» al moverse tecleando SINLAPIZ (SL), CONLAPIZ (CL) volverá a hacer que pinte; GOMA hará, no sólo que no pinte, sino que borre cualquier rastro ya existente por el que pase.

Pero hagamos que la tortuga nos dibuje algo. Con BORRAPANTALLA podemos volver a empezar con la pantalla limpia y la tortuga en el centro. Para no ser menos, haremos lo que todo el mundo cuando empieza a trabajar con Logo: un cuadrado. La forma más prosaica sería:

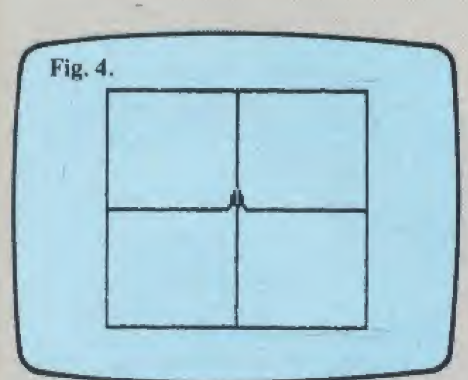
AVANZA 40 GIRADERECHA 90
AVANZA 40 GIRADERECHA 90
AVANZA 40 GIRADERECHA 90
AVANZA 40 GIRADERECHA 90

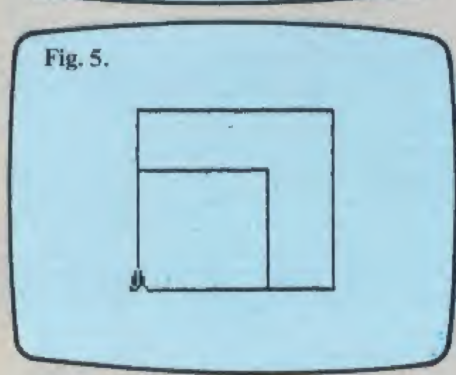
Esta forma es un poco laboriosa, sobre todo si sabemos que a nuestra tortuga le podemos decir que repita algo las veces que queramos. Con REPITE 4 [AVANZA 40 GIRADERECHA 90] obtendremos el mismo resultado: Un cuadrado de 40 pixels de lado.

Pero podemos hacer que la tortuga «recuerde» este conjunto de instrucciones bajo el nombre de CUADRADO (o el que nos apetezca), y así no tendremos que escribir todas las instrucciones cada vez que queramos que nos dibuje un cuadrado. Esto es definir un procedimiento; para hacerlo tecleamos:

? PARA CUADRADO
> REPITE 4 [AVANZA 40 GIRADERECHA 90]
> FIN

Al poner FIN nos aparece el mensaje CUADRADO definido. El vocabulario de la tortuga ha sido ampliado con un nuevo término. Podemos ahora utilizar el procedi-


Fig. 4.


Fig. 5.

miento CUADRADO junto con todas las primitivas para definir nuevos procedimientos. Si tecleamos: REPITE 4 [CUADRADO GIRADERECHA 90] obtenemos la figura 4. A esto le podemos dar un nombre, por ejemplo, reja, y definirlo como un procedimiento.

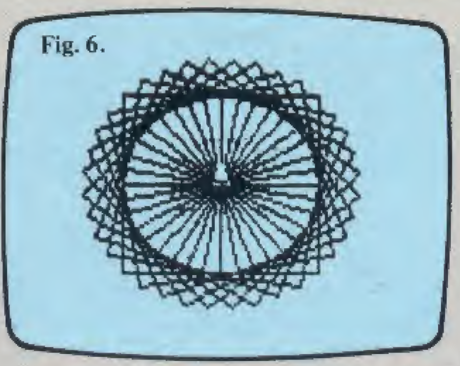
Nuestro cuadrado y nuestra reja tienen limitaciones. Su tamaño es fijo. Si queremos que la tortuga dibuje cuadrados con tamaños diferentes podemos utilizar variables:

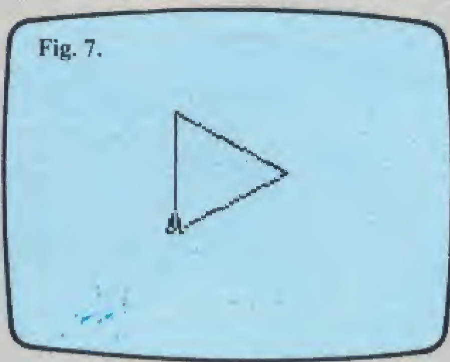
? PARA CUADRADO : lado
> REPITE 4 [AVANZA : lado GIRADERECHA 90]
> FIN

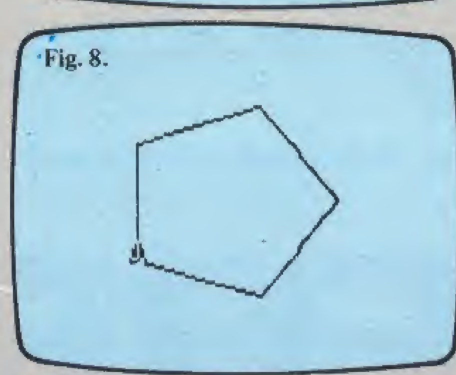
Cada vez que ordenemos CUADRADO tendremos que indicar el tamaño del lado: CUADRADO 10, CUADRADO 20... (véase la figura 5).

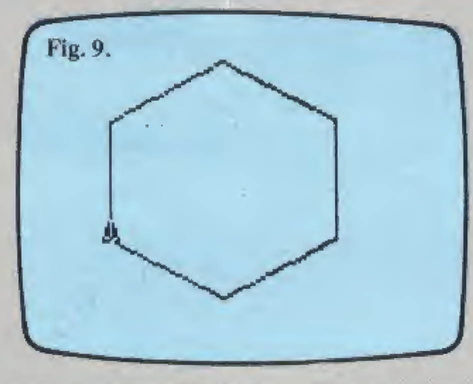
Por el contrario, la instrucción REPITE 36 [CUADRADO 15 GIRADERECHA 10] producirá un bonito gráfico, como se ve en la figura 6.

La variable lado sólo tiene valor al realizar el procedimiento CUADRADO, fuera de él, el Logo no la reconoce; es una variable local. Las variables locales van asociadas


Fig. 6.


Fig. 7.


Fig. 8.


Fig. 9.

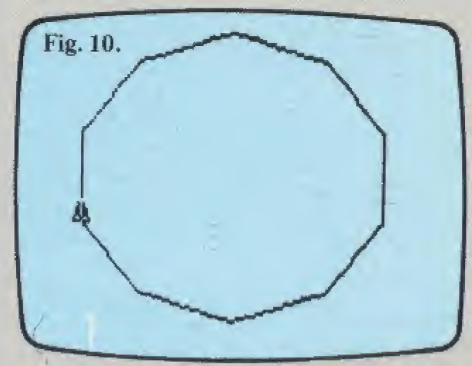
a un procedimiento. En nuestro ejemplo lado a CUADRADO. Si queremos ver el valor del lado después de dibujarlo tecleamos...

IMPRIME : lado

(Para referirse al contenido de una variable hay que poner dos puntos (:) por delante.) Nos aparecerá el mensaje:

lado no tiene valor

Una variable es global, válida para todos los procedimientos, si la creamos con la orden ASIGNA.


Fig. 10.

Veamos un ejemplo:

ASIGNA "lado 15

ASIGNA hará que en cualquier procedimiento que se use la variable lado, ésta tenga el valor 15. Nótese que al asignarle un valor anteponemos comillas (") al nombre de la variable. Los dos puntos (:) indican el contenido de una variable y las comillas (") el nombre. Podemos introducir:

ASIGNA "lado 1 + : lado
IMPRIME : lado

a lo que responderá 16. ¡Intentemos algo menos cuadrado que un CUADRADO! Polígonos. Para ello usaremos lo visto hasta ahora. Un giro completo de la tortuga y de cualquiera que lo haga es de 360°. En cuadrado (¡vaya otra vez!). Cuatro giros de 90° dan los 360°. Pero si hay más lados debemos girar un poco menos. En resumen, el giro será siempre de 360° dividido entre el número de lados. Veamos:

? PARA POLIGONO : lados : tamaño
> REPITE : lados [AVANZA : tamaño GIRADERECHA 360/: lados]
> FIN

Como veis, podemos usar más de una variable local en un procedimiento: lados y tamaño en este caso.

POLIGONO 3 20
POLIGONO 5 20
POLIGONO 6 20
POLIGONO 10 20

nos darían polígonos de 3, 5, 6 y 10 lados respectivamente (ver figuras 7 a 10). Esto cada vez se parecerá más a un círculo. ¿Es un círculo un polígono de infinito número de lados?

Empezar a trabajar en código máquina, es empezar una carrera de obstáculos en la que la "valla" de la interrupciones parece ser la más difícil de rebasar. Al menos, eso se desprende de las dudas que los lectores de TODOSPECTRUM nos vienen planteando. Nuestro "experto" analiza el tema en profundidad.

INTERRUPC
DEL Z-80

El propósito que persigue un circuito periférico al generar una interrupción es que la CPU abandone el programa que la ocupa, a fin de ejecutar un programa distinto, generalmente relacionado con la propia interrupción. Es frecuente que después de ejecutarlo, podamos volver al programa inicial, aunque esto depende del tipo concreto de interrupción. El proceso suele tener marcadas semejanzas con el salto a una subrutina, con la particularidad de que se produce por órdenes externas.

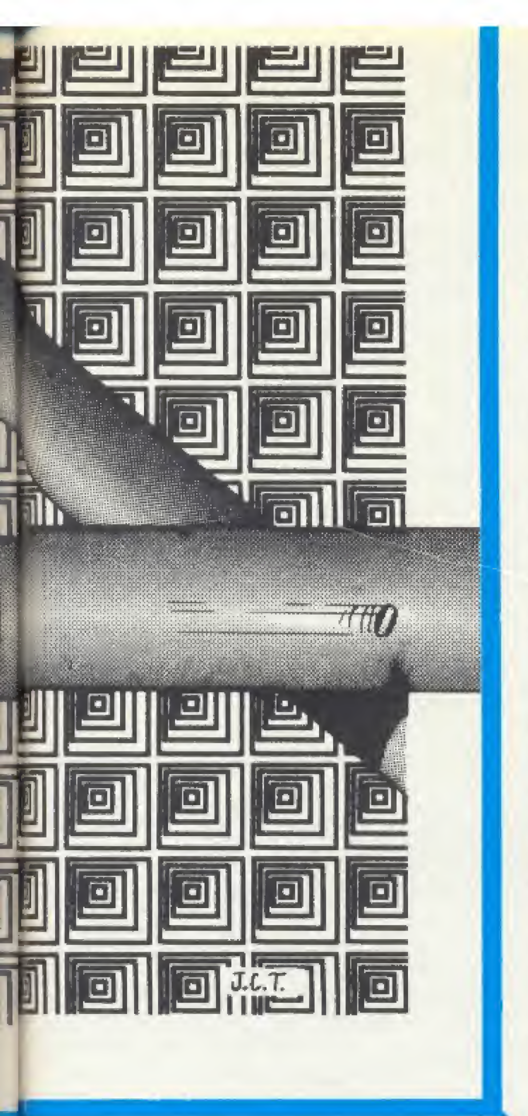
Para explorar las ventajas —incluso la necesidad— de las interrupciones, vamos a poner una serie de ejemplos que aclararán conceptos.

Supongamos en primer lugar que poseemos un gran ordenador que dispone de unidades de disco. Estos dispositivos de memoria se caracterizan por tener una gran capacidad de memoria a precios reducidos, con el inconveniente de que el tiempo de acceso a la información es muy elevado. (Tiempo de acceso: tiempo que transcurre desde que se solicita la información hasta que se dispone de ella). Cuando la CPU necesita unos datos almacenados en disco, no le son enviados inmediatamente, sino después de algunos segundos. Tal vez nos parezca poco tiempo, pero en este intervalo la CPU puede realizar muchos millones de operaciones. Por ello sería un auténtico despilfarro perder este tiempo. Un sistema muy eficaz sería que la unidad de disco indicara

Varios segundos son suficientes para realizar millones de operaciones.

a la CPU que los datos están listos. Para conseguir esto es necesario generar una interrupción.

Para recoger la información procedente del teclado, tenemos dos



Spectrum. Esta CPU dispone de dos patillas para interrupciones. Son la NMI y la INT. Ambas se escriben con una barra encima para indicar que funcionan con lógica negativa, a saber: son inactivas cuando están a nivel lógico 1 y generan la interrupción cuando se las lleva a un estado lógico bajo. Existen dos tipos de interrupciones, las evitables y las no evitables.

Interrupciones no evitables

Cuando un periférico activa la línea NMI (*Non Maskable Interrupt,* interrupción no evitable), la CPU ignora la instrucción que acaba de cargar de la memoria y hace una llamada a la subrutina que se encuentra en la dirección 0066H (supuesto que el programador se haya tomado la molestia de escribirla). Para volver al programa que se dejó, se usará la instrucción RETN.

Este tipo de interrupciones se usa cuando existan asuntos de gran importancia que requieran atención inmediata. El Spectrum no usa la línea NMI.

Interrupciones evitables

La línea INT se encarga de generar interrupciones que pueden ser deshabilitadas por programa. Los que deseen profundizar más en el tema, estudien detenidamente la figura 2.

Los programas donde existan constantes de tiempo (tales como el de la gestión de cinta o en los BEEP) es muy importante que la CPU no se sea interrumpida por tareas de poca importancia. Sin embargo dichos asuntos pueden tener una importancia capital en otros momentos (tales como la lectura de teclado que se realiza a instancias de una interrupción cada 20 mS). Como el periférico no tiene modo de saber si su presencia es deseada o se trata de una intromisión imperdonable, se ha optado porque sea el propio programa quien decida su actitud ante la interrupciones.

Cuando la CPU empieza a funcionar, las interrupciones están deshabilitadas. Consecuentemente toda actuación sobre la línea INT será ignorada. Las interrupciones no evitables se atienden siempre.

Cuando se ejecuta la instrucción EI (*Enable Interrupt,* habilitación de interrupciones), la CPU estará en condiciones de responder a todos los requerimientos que se hagan sobre la línea INT. Si estando en este modo se encuentra la instrucción DI (*Disable Interrupt,* deshabilitación de interrupciones), se hará de nuevo, caso omiso de las interrupciones evitables.

CIONES

Hay dos tipos de interrupciones: las evitables y las no evitables.

opciones. Uno muestrearlo periódicamente, otro, detectar cuando se pulsa una tecla y en este caso avisar al ordenador del evento. En ambos casos, será necesario abandonar el programa en curso a instancias externas para obrar en consecuencia.

Supongamos por último que, a causa de un error de programación, el ordenador está ejecutando un bucle sin salida. ¿Qué hacer? Un Spectrum puede desconectarse, muchas veces con gran dolor, pero esta solución es impensable en un gran ordenador.

Vamos a pasar de un ordenador genérico al microprocesador Z-80 que es el que usa nuestro querido

Modos de interrupción

Existen tres modos de interrupción evitable. Para seleccionar una de ellas, el programador usará las instrucciones IM 1, IM 2 o IM 3 (*Interuption Mode,* modo de interrupción). A lo largo del programa se podrá pasar de unos modos a otros. En todos ellos será posible volver al punto de ruptura usando la instrucción RETI. Pasemos a la descripción de cada uno de ellos.

— MODO 0: Es semejante a las interrupciones del 8080, famoso uP, con el que el Z-80 es compatible en *soft* y *hardware* aunque lo supera en numerosos aspectos.

El circuito que genera la interrupción pone en el *bus* de datos una instrucción y la CPU la ejecuta. Podríamos decir que el circuito "suplanta" a la memoria.

Generalmente la instrucción es un salto a una de las subrutinas en página cero (RST, *ReSTart Routines*), ya que tales instrucciones ocupan un único byte. Sin embargo nada impide que sea una orden de cualquier otro tipo y longitud.

— MODO 1: Es la que usa el Spectrum. La CPU responde a la

Las interrupciones del Spectrum producen un salto a la dirección 0038H.

interrupción haciendo una llamada a subrutina en la dirección 0038H (el proceso es semejante a la respuesta a la NMI, pero dirigiéndose a una dirección diferente).

— MODO 2: Es el modo más potente y ha sido pensado para usarse con los periféricos del Z-80, como el Z-80 PIO (*Programable Input Output*, puerto programable de entrada-salida) y el Z-80 CTC (*Counter-Timer Circuit*, circuito contador-temporizador).

Debe existir en algún lugar de la memoria una tabla que contenga las direcciones (en grupos de 2 bytes) de las rutinas de servicio. Cuando se produce una interrupción, formaremos un número de 16 bits. La parte baja de este número la suministra el periférico que genera la interrupción. La parte alta será el contenido del registro I, que habrá sido previamente cargado con la instrucción LD I, A. El número obtenido es una dirección de memoria, que apunta a uno de los elementos de la tabla. La CPU recogerá el contenido de este punto y del siguiente. Serán respectivamente la parte baja y alta de la dirección de la rutina de servicio de la interrupción.

Así diferentes periféricos darán diferentes datos con los que obtendremos también diferentes direcciones de salto, permitiendo que cada interrupción sea atendida por su subrutina correspondiente.

Para aclarar conceptos y de paso para responder a las cartas de numerosos lectores, vamos a tomar como ejemplo la rutina de RESET que apareció en el número 2 de TODOSPECTRUM. En el mencionado artículo, el registro I guardaba el valor 80H. Como el Spectrum está diseñado para trabajar siempre en modo de interrupción 1, los circuitos periféricos (la ULA es el único que tiene) no hace nada cuando se produce una interrupción, no pone ningún dato en el *bus*, lo que se interpreta como si todos los bits estuvieran a 1, es decir el dato leído es el FFH. La dirección resultante será la 80FFH. En esta dirección IADDR, está contenido el valor 01 81. La CPU realizará entonces una llamada a la subrutina en la dirección 8110H, (la etiquetada como TEST RESET), donde se comprueba si se han pulsado las teclas correspondientes. Como se deduce del contexto el programa no es relocalizable.

Algunos lectores nos han comunicado que esta rutina plantea problemas cuando se tiene el *interface Joystick*. Incluso en algunos casos se llegaba a borrar la memoria. En efecto este hecho es fácilmente comprensible y debe imputarse al diseño poco cuidadoso de los *interfaces* mencionados.

Luis Miguel

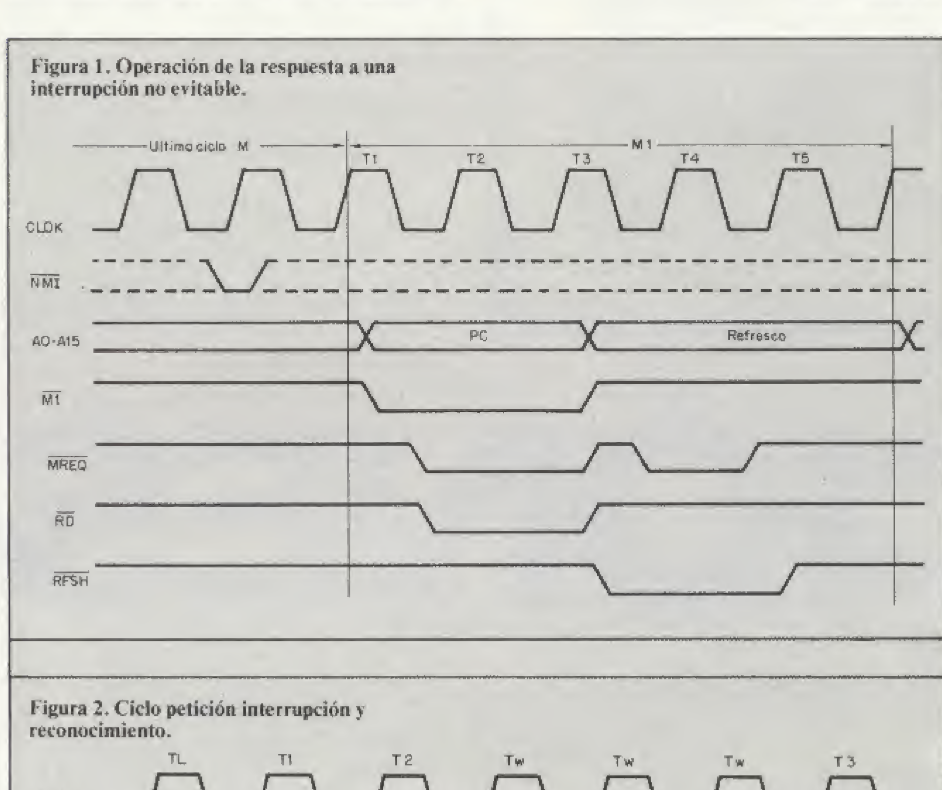


Figura 1. Operación de la respuesta a una interrupción no evitable.

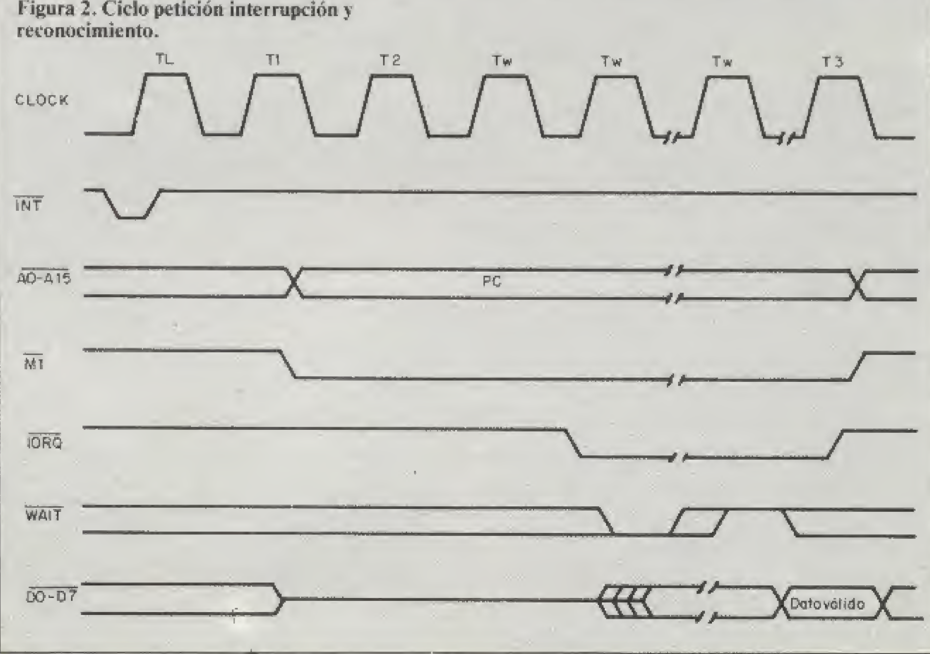


Figura 2. Ciclo petición interrupción y reconocimiento.

AUN ESTAS A TIEMPO **GRAN FINAL:**

11 DE MAYO DE 1985

HEMOS RETRASADO LA GRAN FINAL
Envía rápidamente tu tarjeta de participación.



Diviértete consiguiendo regalos tan estupendos como los de la "tele"

©RTVE



¡Busca tu Chollo! ¡Participa y gana!



Concursa en casa con los protagonistas de la "tele".

Con tu Spectrum (no importa el modelo, 16 ó 48 K.) y la cinta del "Un, Dos, Tres..." podrás conseguir diversión sin límites junto con los tuyos.
Por primera vez se te ofrece un concurso participativo para microordenador, donde pueden competir hasta tres parejas. Y con la novedad de que el juego será cada vez distinto, pues siempre que conectes la cinta al Spectrum, te ofrecerá diferentes alternativas.

UN JUEGO PARA DISFRUTAR CON TUS AMIGOS

Bigote Arrocet te invita en persona a un concurso exactamente igual que el de la "tele". Basta con que pongas la cinta en tu cassette y él te irá dando las instrucciones de este nuevo y diferente juego. Puedes conseguir premios tan sorprendentes como los del programa de Chicho y, por supuesto; también tiene Chollo.

1 de cada 3 cassetes contiene ¡PREMIO DIRECTO!:
Viajes, moto-vespas, sintetizadores, microordenadores Spectrum, QL, un montón de premios más, **y el gran Chollo.**

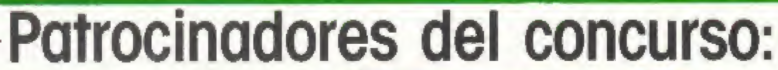


J. M. PUBLICIDAD

¡TRES MILLONES DE PESETAS!

en premios que puedes conseguir participando en la gran final.
No pierdas más tiempo, compra la cinta del "Un, Dos, Tres!













MOTO VESPA

BELLTONS



Información y venta exclusiva en la red de Concesionarios Autorizados INVESTRONICA

EN EL PROXIMO NUMERO DE ESTA REVISTA
¡¡Relación de premios!!

Juego de los espejos

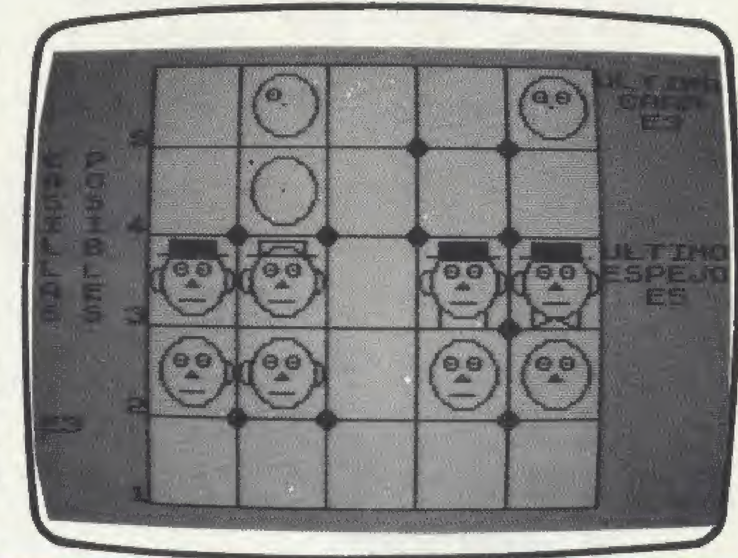
31

Este juego de los espejos esta basado en el cuento hindú "Acercamiento al Al-mutasim". El cuento narra la búsqueda del alma de Al-mutasim a través de los reflejos que dejaba en otros personajes. De forma similar, el juego ha llevado esta búsqueda a la pantalla del televisor, en una cuadrícula de 5 × 5 posiciones sobre la que se va formando la cara de Al-mutasin según la selección de espejos que se realice. Cuando ya se han formado todos sus rasgos, es decir, cuando el reflejo frente a un espejo no aporta nada nuevo al rostro del hindú (o de quien su imaginación disponga) habra finalizado la partida y su posibilidad de ganar al ordenador.

No es necesario entrar en detalle sobre las instrucciones de manejo: se ofrece completa información por pantalla, incluida una demostración de cómo jugar.

Aunque los ritos del mundo oriental sean un poco diferentes, la trascendental búsqueda del alma es también preocupación de occidente ¿no le parece?

Autor: Matías Vaquero — 16K




```
  50 RUN 1060
  60 DATA 48,80,112,144,176,208
  70 DATA 8,40,72,104,136,168
  80 REM
  90 REM            CUADRICULA
 100 REM
 110 CLS : BORDER 4: PAPER 4: IN
K 1: CLS
 115 FOR f=1 TO 20: PRINT AT f,6
: PAPER 5;"                    "
: NEXT f
 120 PRINT AT 1,26;"ULTIMA";AT 2
,26 ;" CARA ";AT 9,26;"ULTIMO";A
T 10,26;"ESPEJO"
 130 RESTORE : FOR f=1 TO 6: REA
D x: PLOT x,8: DRAW 0,160: NEXT
f
 140 FOR f=1 TO 6: READ y: PLOT
48,y: DRAW 160,0: NEXT f
 150 PRINT AT 21,6;"A   B   C
D   E"
 160 PRINT AT 20,5;"1";AT 16,5;"
2";AT 12,5;"3";AT 8,5;"4";AT 4,5
;"5"
 170 RETURN
 180 REM
 190 REM        COORDENADAS DEL
 200 REM               PUNTO
 210 REM
 220 PRINT AT 14,27; PAPER 2; IN
K 6; FLASH 1;"JUEGA";AT 15,27;"E
L  ";t: BEEP .5,15: PRINT AT 14,
27;"JUEGA";AT 15,27;"EL  ";t
 230 IF mov=1 THEN  RETURN
 240 GO SUB borrado
 250 INPUT "Coordenadas del espe
jo?";p$
 260 IF LEN p$<>2 THEN  PRINT #1
; PAPER 2; INK 6; FLASH 1;"PONGA
 LA LETRA SEGUIDA DEL NUM.": BEE
P 1,-10: GO TO 250
 270 IF CODE p$(1)<66 OR CODE p$
(1)>69 OR CODE p$(2)<50 OR CODE
p$(2)>53 THEN  PRINT #1; PAPER 2
; INK 6; FLASH 1;" DEBE ESTAR D
ENTRO DEL TABLERO ": BEEP 1,-10:
 GO TO 250
 280 LET yp=y(VAL p$(2))
 290 LET xp=x(CODE p$(1)-64)
 300 IF POINT (xp+1,yp+1) THEN
PRINT #1; PAPER 2; INK 6; FLASH
1;"       AHI YA HAY UN ESPEJO
   ": BEEP 1,-10: GO TO 250
 310 FOR f=1 TO 3: CIRCLE xp,yp,
f: NEXT f
 320 PRINT AT 11,28;p$
 330 REM  condicion de reflexion
 340 LET xi1=xc-16
```

Programas

```
 350 FOR f=1 TO 6
 360 IF xi1=x(f) THEN  LET vi=f:
 GO TO 380
 370 NEXT f
 380 LET vi=vi+64: LET v$=CHR$ v
i
 390 LET yi1=yc+2*(yp-yc)-16
 400 FOR f=1 TO 6
 410 IF yi1=y(f) THEN  LET vi=f:
 GO TO 430
 420 NEXT f
 430 LET xi2=xc+2*(xp-xc)-16
 440 FOR f=1 TO 6
 450 IF xi2=x(f) THEN  LET w=f:
GO TO 470
 460'NEXT f
 470 LET w=w+64: LET w$=CHR$ w
 480 LET yi2=VAL c$(2)
 490 REM ████casillas posibles██
 500 LET a$="CASILLAS": LET g$="
POSIBLES"
 510 FOR f=1 TO 8: PRINT AT f+4,
1;a$(f): BEEP .01,f*2: PRINT AT
f+4,3;g$(f): BEEP .01,f*2+1: NEX
T f
 520 PRINT AT 15,1;v$;vi;AT 17,1
;w$;yi2
 530 IF CODE v$<65 OR CODE v$>69
 THEN  PRINT AT 15,0;"       "
 540 IF CODE w$<65 OR CODE w$>69
 THEN  PRINT AT 17,0;"       "
 550 IF vi<1 OR vi>5 THEN  PRINT
 AT 15,0;"       "
 560 IF yi2<1 OR yi2>5 THEN  PRI
NT AT 17,0;"       "
 570 IF xi1<46 OR xi1>176 OR yi1
<8 OR yi1>136 THEN  GO TO 590
 580 IF POINT (xi1+16,yi1+16) TH
EN  PRINT AT 15,0;"       "
 590 IF xi2<48 OR xi2>176 OR yc<
24 OR yc>152 THEN  GO TO 610
 600 IF POINT (xi2+16,yc) THEN
PRINT AT 17,0;"       "
 610 LET h$=SCREEN$ (15,1): LET
j$=SCREEN$ (15,2): LET k$=SCREEN
$ (17,1): LET l$=SCREEN$ (17,2)
 620 LET gan=t+1: IF gan=3 THEN
 LET gan=1
 630 IF h$+j$+k$+l$="    " THEN
 PRINT #1; PAPER 6: INK 2;" GANA
DOR EL ";gan;". OTRA PARTIDA?":
PAUSE 0: GO TO CODE INKEY$*100
 640 RETURN
 650 REM ██████████████████████
 660 REM ████████COORDENADAS ███
 670 REM ████████DE LA CASILLA██
 680 REM ██████████████████████
 690 INPUT "Coordenadas de la ca
silla?";c$
 700 IF LEN c$<>2 THEN  GO TO 74
0
 710 IF mov=1 THEN  GO TO 730
 720 IF c$<>h$+j$ AND c$<>k$+l$
THEN  GO TO 750
 730 IF CODE c$(1)>=65 AND CODE
c$(1)<=69 AND CODE c$(2)>=49 AND
 CODE c$(2)<=53 THEN  GO TO 760
 740 PRINT #1; PAPER 2; INK 6; F
LASH 1;"TECLEE LA LETRA SEGUIDA
DEL NUM.": BEEP 1,-10: GO TO coo
rdc
 750 PRINT #1; FLASH 1; PAPER 2;
 INK 6;"        AHI NO PUEDE PONE
R        ": BEEP 1,-10: GO TO coo
rdc
 760 PRINT AT 3,28;c$: DIM x(6):
 DIM y(6)
 770 RESTORE : FOR f=1 TO 6: REA
D x(f): NEXT f
 780 FOR f=1 TO 6: READ y(f): NE
XT f
 790 LET xc=x(CODE c$(1)-64)+16
 800 LET yc=y(VAL c$(2))+16
 820 REM ███████cara███████
 830 PLOT xc,yc: CIRCLE xc,yc,12
 840 IF mov>1 THEN  PLOT xc-4,yc
+4: CIRCLE xc-4,yc+4,2
 850 IF mov>2 THEN  PLOT xc+4,yc
+4: CIRCLE xc+4,yc+4,2
 860 IF mov>3 THEN  PLOT xc,yc:
DRAW -2,-2: DRAW 4,0: DRAW -2,2
 870 IF mov>4 THEN  PLOT xc-4,yc
-7: DRAW 8,0
 880 IF mov>5 THEN  PLOT xc+11,y
c+4: DRAW 0,-8,-PI
 890 IF mov>6 THEN  PLOT xc-11,y
c+4: DRAW 0,-8,PI
 900 IF mov>7 THEN  PLOT xc-12,y
```

```
c+9: DRAW 24,0: PLOT xc-8,yc+8:
DRAW 0,6: DRAW 16,0: DRAW 0,-6
 910 IF mov>8 THEN  FOR f=yc+8 T
O yc+14: PLOT xc-8,f: DRAW 16,0:
 NEXT f
 920 IF mov>9 THEN  PLOT xc+8,yc
-9: DRAW 0,-7: PLOT xc-8,yc-9: D
RAW 0,-7
 930 IF mov>10 THEN  DRAW 8,4: D
RAW 8,-4
 940 IF mov>11 THEN  FOR f=1 TO
4: PLOT xc-2,yc-12-f: DRAW 4,0:
NEXT f
 960 RETURN
 970 REM ████████████████████████
 980 REM ████████BORRADO█████████
 990 REM ████████████████████████
1000 FOR f=0 TO 21: PRINT AT f,0
;"        ": NEXT f: RETURN
1010 REM ███████████████████████
1020 REM ████ROTULACION████
1030 REM ███████████████████████
1040 FOR f=1 TO LEN a$: PRINT AT
 1,INT ((31-LEN a$)/2)+f;a$(f):
BEEP .01,RND*20: NEXT f: RETURN
1050 REM ████████████████████████
1060 REM ████INICIALIZACION█████
1070 REM ████████████████████████
1080 POKE 23658,8
1090 LET cuadr=90
1100 LET coordp=190
1110 LET coordc=660
1120 LET borrado=980
1130 LET rot=1020
1140 LET mov=1: LET t=0
1150 REM ████████████████████████
1160 REM ████INSTRUCCIONES█████
1170 REM ████████████████████████
1175 CLS : BORDER 6: PAPER 6: IN
K 1
1180 CLS : LET l=10: LET a$="PAR
E LA CINTA": GO SUB ROT
1190 LET L=5: LET A$="EL JUEGO D
E LOS ESPEJOS": GO SUB ROT: LET
L=6: LET A$="QUE SE DESPLAZAN":
GO SUB ROT: LET L=8: LET A$= M
ATIAS VAGUERO 1984": GO SUB ROT
1200 LET l=12: LET a$="QUIERE IN
STRUCCIONES?": GO SUB rot: PAUSE
 0
1210 IF INKEY$="N" THEN  GO TO 1
440
1220 CLS : LET l=1: LET a$="ESTE
 JUEGO ESTA BASADO": GO SUB ROT:
 LET L=2: LET A$="EN EL CUENTO":
 GO SUB ROT: LET L=3: LET A$="''
ACERCAMIENTO A AL-MU'TASIM''": G
O SUB ROT
1230 LET L=4: LET A$="DEL ABOGAD
O HINDU": GO SUB ROT: LET L=5: L
ET A$="MIR BAHADUR ALI": GO SUB
ROT
1240 LET L=6: LET A$="QUE LO SUB
TITULO:": GO SUB ROT: LET L=7: L
ET A$="''UN JUEGO CON ESPEJOS":
GO SUB ROT: LET L=8: LET A$="QUE
 SE DESPLAZAN''": GO SUB ROT
1250 PAUSE 100: LET L=10: LET A$
="APARECE COMENTADO EN LA REVIST
A": GO SUB ROT: LET L=11: LET A$
="'CACUMEN' NUMERO 20": GO SUB R
OT
1260 PAUSE 100: LET L=13: LET A$
="EL CUENTO (IGUAL QUE EL JUEGO)
": GO SUB ROT: LET L=14: LET A$=
"TRATA DE LA BUSQUEDA DEL ALMA":
 GO SUB ROT: LET L=15: LET A$="D
E ALMOTASIM A TRAVES DE LOS": GO
 SUB ROT: LET L=16: LET A$="REFL
EJOS QUE DEJA EN OTROS": GO SUB
ROT: LET L=17: LET A$="PERSONAJE
S": GO SUB ROT
1270 PRINT #1;"   PULSE UNA TECL
A PARA SEGUIR  ": PAUSE 0: CLS :
 LET L=1: LET A$="EN CADA PASO S
E FORMA UNA CARA": GO SUB ROT: L
ET L=2: LET A$="(UN POCO OCCIDEN
TALIZADA)": GO SUB ROT: LET L=3:
 LET A$="QUE ES EL REFLEJO DE OT
RA": GO SUB ROT: LET L=4: LET A$
="ANTERIOR": GO SUB ROT
1280 LET L=5: LET A$="HASTA QUE
LLEGA EL MOMENTO": GO SUB ROT: L
ET L=6: LET A$="EN QUE UN ROSTRO
 NO SERA REFLEJO": GO SUB ROT: L
ET L=7: LET A$="DE NINGUN OTRO:"
: GO SUB ROT: LET L=8: LET A$="E
S ALMOTASIM": GO SUB ROT
1290 PAUSE 50: LET L=10: LET A$=
"LA PARTIDA Y LA NOVELA": GO SUB
 ROT: LET L=11: LET A$="HAN TERM
INADO": GO SUB ROT
1300 PAUSE 50: LET L=13: LET A$=
"PARA JUGAR SE DEBERA SITUAR": G
O SUB ROT: LET L=14: LET A$="PRI
MERO UN ESPEJO EN LAS": GO SUB R
```

Aquí el ordenador Hit-Bit de Sony.

Aquí la familia.

Aquí a su izquierda tiene el nuevo ordenador personal Hit-Bit de SONY. Algo especial, el auténtico ordenador doméstico. Repetimos, es de SONY.

A la derecha tenemos a una familia. Normal. Como la suya o la de tantos. Con problemas o no, con aficiones y con ganas de tenerlo todo muy bien ordenado.

El hombre puede usar el Hit-Bit para resolver sus asuntos profesionales a la perfección.

Pero también en casa Hit-Bit echa una mano: contabilidad del hogar, agenda familiar y todo lo que haya que ordenar.

Y todos los comecocos, marcianitos y monstruitos que su hijo le pida. Pero también una amplia gama de posibilidades en programas educativos.

El Hit-Bit, le ofrece además el Sistema MSX compatible con más de 20 marcas distintas.

También un sistema de notas musicales que le permite crear sus propios efectos o componer una partitura.

Pero aún hay más, el Hit-Bit le ofrece no tan sólo la posibilidad de crear y realizar gráficos, si no que dispone de toda una serie completa de periféricos para que su ordenador se convierta en algo realmente serio. Sólo Sony puede ofrecer en un ordenador de este tipo tantas posibilidades.

Sin compromiso alguno. En cualquier distribuidor SONY pueden presentarse mutuamente. Seguro que se entienden, piense que el Hit-Bit es de SONY. ¿Se empieza ya a imaginar lo que es capaz de hacer?

Hit-Bit. Ya sabe, para lo que Vd. y su familia gusten ordenar.



PRN-C41 IMPRESORA- PLOTTER EN COLOR.

La PRN-C41 le permite imprimir una amplia gama de gráficos utilizando el HIT BIT. Permite utilizar hojas de papel o un rollo continuo, y el texto y gráficos pueden ser escritos y diseñados en negro, azul, rojo o verde. La impresora es ligera y compacta, con un diseño moderno, práctico y atractivo.

HBD-50 MICRO FLOPPYDISK DRIVE.

El HBD-50 se conecta fácilmente al HIT BIT. Diseñado para utilizar los Micro Floppy Disk de 3,5 pulgadas de SONY.





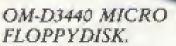
JS-55 MANDO PARA JUEGOS.

Diseñado especialmente para ser utilizado por diestros o zurdos, su manejo es sencillo y su apariencia sumamente atractiva.

EL CARTUCHO HBI-55 LE PERMITE ALMACENAR 4 KBYTES DE INFORMACION PERSONAL.

Gracias a la batería incorporada el HBI-55 guarda los datos aunque se desconecte el ordenador y se extraiga el cartucho.





HBM-16 y HBM-64 CARTUCHOS DE AMPLIACION DE MEMORIA.

Insertando el HBM-16 obtendrá 16 Kbytes extra de memoria RAM. El HBM-64 le ofrece 64 Kbytes

OM-D3440 MICRO FLOPPYDISK.

500 Kbytes de información (más de 500.000 caracteres) caben en estos pequeños diskettes de 3,5 pulgadas. Además, su carcasa protectora le garantiza una larga vida.

TAGG

```
OT: LET L=15: LET A$="INTERSECCI
ONES DE LAS CASILLAS": GO SUB RO
T
1310 LET L=16: LET A$="Y DESPUES
 EL ROSTRO REFLEJADO": GO SUB RO
T: LET L=17: LET A$="DEL ANTERIO
R": GO SUB ROT: PAUSE 50: LET L=
19: LET A$="PIERDE EL QUE NO PUE
DA HACERLO": GO SUB ROT
1320 PRINT #1;"            PULSE UN
A TECLA              PARA DEMOS
TRACCION        ": PAUSE 0
1330 REM ▓▓▓▓▓demostraccion▓▓▓▓▓
1340 LET r$=" B4B5E5E2D2A2B2B3A3
D3E3"
1350 LET q$="   D5D4E4E2C2B2E3B4
C4E5"
1360 GO SUB cuadr
1370 LET t=t+1
1380 IF mov>1 THEN  LET p$=q$(mo
v*2)+q$(mov*2+1): PRINT : INVERS
E 1;AT 15,27;"PULSE";AT 16,27;"
UNA ";AT 17,27;"TECLA": PAUSE 0:
 FOR f=15 TO 17: PRINT AT f,27;"
     ": NEXT f: GO SUB borrado:
GO SUB 260
1390 LET c$=r$(mov*2)+r$(mov*2+1
): GO SUB 700
1400 LET mov=mov+1
1410 IF mov=12 THEN  PRINT #1;"E
L QUE JUEGA PIERDE POR NO PODERC
OLOCAR OTRA CARA.              "
: PAUSE 0: GO TO 1460
1420 GO TO 1370
1430 REM ▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓
1440 REM ▓▓PROGRAMA PRINCIPAL▓▓
1450 REM ▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓
1460 LET mov=1: LET t=1
1470 GO SUB cuadr
1480 GO SUB coordp
1490 GO SUB coordc
1500 LET mov=mov+1: LET t=t+1
1510 IF t=3 THEN  LET t=1
1520 GO TO 1480
1530 REM ▓▓▓▓▓otra o final▓▓▓▓
7800 CLS : PRINT AT 10,10;"HASTA
 OTRA": STOP
8300 GO TO 1440
```

PARA PERFECCIONISTAS

Para los que tengan el programa "SUPERCODE" es aconsejable incorporar las rutinas 79 y 43 para una presentación más vistosa y divertida. Para ello añadir en la 1080 CLS: PRINT AT 10,5; "CARGANRO CODIGO MAQUINA"; AT 12,6; "UN MOMENTO POR FAVOR": LOAD "SCIFI CHAR" CODE 57344,768: LOAD "UNI-NOTE S" CODE 64647,28: POKE 23658,8: POKE 23607,223. (Lógicamente, habrá de introducir estas rutinas a partir de la dirección 57344 y 64647 respectivamente).

Este último poke es para la impresión de los caracteres de una tipografía especial. Si se desea volver a los caracteres normales teclear POKE 23607,60.

"UNI-NOTE S" necesita de 4 POKEs anteas de hacer el RANDOMIZE POKE 64648,fr: POKE 64649,sp: POKE 64651,du: POKE 64670,h. Añadiendo a esto RANDOMIZE USR 64647; RETURN tenemos una rutina que podemos llamar "son". En cualquier parte del programa donde normalmente llamamos la atención con un BEEP, se pueden dar valores a (fr)ecuencias, (sp)an, (du)ración; si h=28 el tono descenderá, mientras que si h=29 será ascendente.

Haga pruebas si quiere o introduzca estos valores: sp=100.

LINEAS	CONCEPTO	fr	du	h
220	"JUEGA EL..."	100	50	29
250, 690	INPUT P$ y c$	100	10	29
260, 300, 740, 750	Errores	50	30	28
630	"GANADOR EL..."	50	5	28
1380	"PULSE UNA TECLA"	100	20	29
1380	Antes de GOSUB borrado	50	30	28

Añadir después GOSUB son, y en el 4.° caso igual pero dentro de un bucle FOR f=1 to 10: GOSUB son: NEXT f

Intelect

No le vamos a decir que con este programa su ordenador pasará a ser "inteligente" o que con él comienza la era de "inteligencia artificial" en el Spectrum. Lo que sí le diremos es que pasará un rato divertido con las contestaciones que le dará su ordenador.

Se trata, en definitiva, de dotar al ordenador de la posibilidad de aprender y enseñar las diferencias entre un cierto grupo de elementos comunes entre sí. El método empleado se basa en una pregunta directa con dos posibles soluciones (sí/no), que dará paso después a dos soluciones indirectas. El programa trabaja con el ejemplo de las provincias españolas, de fácil traslación hacia cualquier otro tema, modificando al sentencia 100. El esquema se muestra en la figura 1. El ordenador elije la respuesta y pregunta si es correcta. En caso negativo, pregunta cuál es esta y en qué se diferencia de las demás, guardándola en memoria (Ver figura 2. Los datos introducidos son VALENCIA y TIENE NARANJAS). Así, la respuesta BARCELONA se convierte en una nueva pregunta ¿tiene naranjas?, que a su vez se acepta dos soluciones: SI = Valencia y NO = Barcelona. ¡El ordenador está aprendiendo!

Más tarde, cuando todos los datos están diferenciados, el programa puede aplicarse a la enseñanza, por ejemplo dando pistas al ordenador para que adivine un elemento pensado de antemano.

El problema de grabación de datos se resuelve mediante dos tablas. Una tabla numérica de dos elementos almacena el subíndice de otra alfanumérica donde se encuentra la pregunta siguiente o la respuesta conseguida. Obviamente, la alfanumérica almacena esas preguntas y respuestas. (La figura 3 ilustra un esquema de la tabla). Si la respuesta a la primera pregunta (¿ESTA EN LA PENINSULA?) es afirmativa, se presenta en pantalla la segunda respuesta. Si es negativa, se presenta la respuesta tercera. Al introducir un nuevo dato, se varía la información (ver figura 4). Ahora la segunda respuesta se ha

Figura 1.

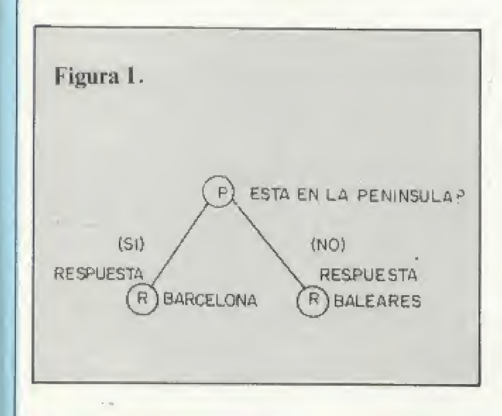
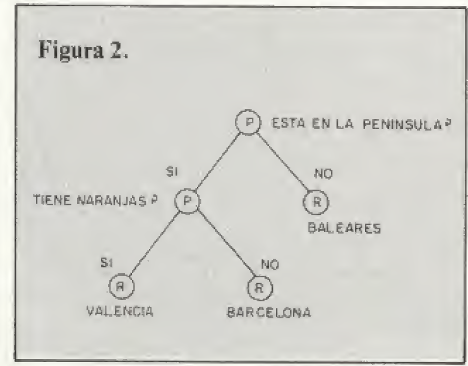


Figura 2.



convertido en la segunda pregunta con dos nuevas respuestas, la cuarta y la quinta.

Respuestas y preguntas quedan bien diferenciadas si se observa que las respuestas equivale en la tabla numérica a los valres 0,0 mientras que las preguntas equivalen a dos números diferenciados de 0.

Autor: Andres Martínez 16K

Figura 3.

SI	NO		
2	3	1	ESTA EN LA PENINSULA
4	5	2	TIENE NARANJAS
0	0	3	BALEARES
0	0	4	VALENCIA
0	0	5	BARCELONA

Figura 4.

SI	NO		
2	3	1	ESTA EN LA PENINSULA
0	0	2	BARCELONA
0	0	3	BALEARES
		4	

```
  3 REM ***********************
  4 REM
  5 REM      "INTELECT"
  6 REM ANDREU MARTINEZ SUAU
  7 REM     ANDREU 1984
  8 REM
  9 REM ***********************
 10 LET sw=0
 20 PRINT "Es la primera vez qu
e usas"'"este programa? (s/n)"
 30 PAUSE 0: IF INKEY$="s" THEN
 GO TO 80
 40 IF INKEY$<>"n" THEN  GO TO
30
 50 PRINT ''"Te interesa cargar
 algun dato"'"de la cinta? (s/n)
"
 60 PAUSE 0: IF INKEY$="n" THEN
 GO TO 80
 70 IF INKEY$<>"s" THEN  GO TO
60
 75 LET sw=1
 80 DIM a(100,2): DIM a$(100,30
)
 90 IF sw=1 THEN  PRINT ''"Pues
 prepara la cinta y "'"pulsa PLA
Y": PAUSE 250: LOAD "a()" DATA a
(): LOAD "a$()" DATA a$(): GO TO
 110
 100 LET a$(1)="Esta en la penin
sula": LET a$(2)="barcelona": LE
T a$(3)="baleares": LET a(1,1)=2
: LET a(1,2)=3
 110 REM averiguar
 120 CLS : LET x=1
 130 GO SUB 430: PRINT ''a$(x, T
O f);"? (s/n)"
 140 PAUSE 0: IF INKEY$="n" THEN
 LET x=a(x,2): GO TO 170
 150 IF INKEY$<>"s" THEN  GO TO
140
 160 LET x=a(x,1)
 170 IF a(x,1)=0 THEN  GO TO 190
 180 GO TO 130
 190 REM posibilidad
 200 GO SUB 430: PRINT ''"Es ";a
$(x, TO f)
 210 PRINT ''"Estas de acuerdo?
(s/n)"
 220 PAUSE 0: IF INKEY$="s" THEN
 CLS : PRINT AT 10,9;"LA HE ACE
RTADO";AT 11,9;"SOY UN GENIO": P
AUSE 250: CLS : GO TO 325
 230 IF INKEY$<>"n" THEN  GO TO
220
 235 REM bucle posicionar resp.
 240 FOR b=1 TO 100
```

```
Es la primera vez que usas
este programa? (s/n)

Esta en la peninsula? (s/n)

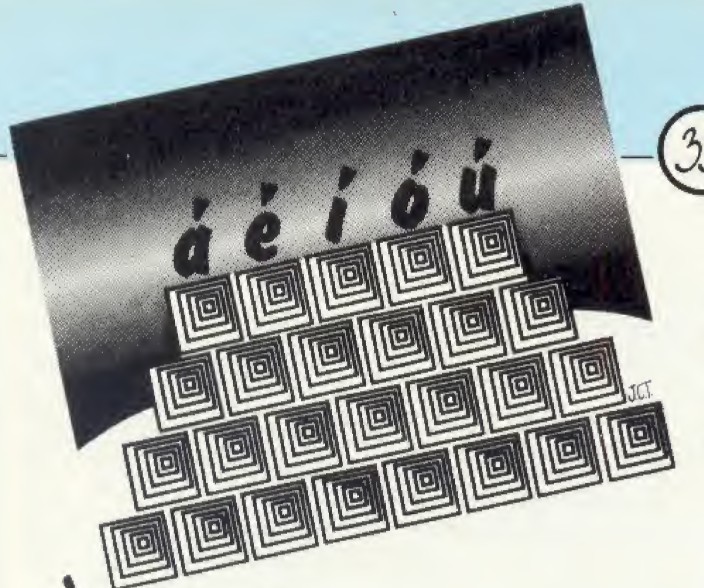
Es barcelona

Estas de acuerdo? (s/n)

Gracias por el dato.

Qieres seguir? (s/n)
```

```
 250 IF a$(b,1)=" " THEN  GO TO
280
 260 NEXT b
 270 PRINT ''"Lo siento, ya se d
emasiado": PAUSE 200: GO TO 350
 280 INPUT "Cual es? ";a$(b)
 290 LET a$(b+1)=a$(x)
 295 GO SUB 470
 300 INPUT "cual es la diferenci
a entre ";(a$(b, TO z));" y ";(
a$(b+1, TO f));"?";'(a$(b, TO z)
);a$(x)
 310 LET a(x,1)=b: LET a(x,2)=b+
1
 320 PRINT ''"Gracias por el dat
o."
 325 PRINT ''"Qieres seguir? (s/
n)"
 330 PAUSE 0: IF INKEY$="s" THEN
  GO TO 120
 340 IF INKEY$<>"n" THEN  GO TO
330
 350 REM grabar los datos
 360 CLS : PRINT "Quieres grabar
 los datos que se?(s/n)"
 370 PAUSE 0: IF INKEY$="n" THEN
  STOP
 380 IF INKEY$<>"s" THEN  GO TO
370
 390 PRINT ''"Prepara la cinta":
 PAUSE 250: SAVE "a()" DATA a():
 SAVE "a$()" DATA a$()
 400 CLS : PRINT "Rebobina la ci
nta para"'"verificarlos": PAUSE
250: VERIFY "a()" DATA a(): VERI
FY "a$()" DATA a$()
 410 STOP
 430 REM cortar a$
 440 FOR f=LEN a$(x) TO 1 STEP -
1
 450 IF a$(x,f)=" " THEN  NEXT f
 460 RETURN
 470 FOR z=LEN a$(b) TO 1 STEP -
1
 480 IF a$(b,z)=" " THEN  NEXT z
 490 RETURN
```



33

Vocales acentuadas

Afortunadamente con el ordenador no hay problemas con los acentos: siempre se le puede echar la culpa que no tiene. Pero si quiere ser riguroso, el ejército le echa una mano, o al menos el Teniente Coronel Castillo, quien también dispone de un Spectrum. Además de las vocales, con este pequeño programa podrá obtener también el signo de interrogación de admiración. La rutina propiamente dicha va de las instrucciones 9000 a 9180.

TECLAS

Caracter	Resultado	Teclas
— a acentuada:	á	A
— e acentuada:	é	E
— i acentuada:	í	I
— o acentuada:	ó	O
— u acentuada:	ú	U
— letra ñ:		N
— letra Ñ:		M
— º de 1.º:		O
— ª de 1.ª:		P
— interrogación:	¿	C
— admiración:	!	H

Autor: Rodolfo Castillo — 16 K

```
¡Vaya! Una rutina para textos
en Español. Ejemplo práctico:
  "En 1ª línea de playa y en un
2º piso pasamos el último vera-
no,pero añorando el campo pró-
ximo. ¿Qué tendrá la campiña?.
NIÑOS y MAYORES la ansían."
   Teclas para estos gráficos:
   ===========================
    A->■ E->■ I->■ O->■ U->■
N->■ M->■ O->■ P->■ C->■ H->■

Program finished, 100:3
```

```
  10 REM Autor: Rodolfo Castillo
©.ZX Spectrum 16/48 K. 1984
  15 REM Programa para caractere
s españoles"
  20 GO SUB 9100
  30 REM Programa muestra en BAS
IC
  40 CLS : PRINT AT 1,1;"¡Vaya!
Una rutina para textos"''" en Es
pañol. Ejemplo práctico:"
  50 PRINT AT 5,2;" ""En 1ª líne
a de playa y en un"''" 2º piso p
asamos el último vera-"''" no,pe
ro añorando el campo pró-"''" xi
mo. ¿Qué tendrá la campiña?."
  60 PRINT AT 13,1;"NIÑOS y MAYO
RES la ansían."""
  70 PRINT AT 15,3;"Teclas para
estos gráficos:";AT 16,3;"======
======================"
  80 PRINT AT 18,4;"A->"; FLASH
1;"á"; FLASH 0;AT 18,9;"E->"; FL
ASH 1;"é"; FLASH 0;AT 18,14;"I->
"; FLASH 1;"í"; FLASH 0;AT 18,19
;"O->"; FLASH 1;"ó"; FLASH 0;AT
18,24;"U->"; FLASH 1;"ú"; FLASH
0
  90 PRINT AT 20,1;"N->"; FLASH
1;"ñ"; FLASH 0;AT 20,6;"M->"; FL
ASH 1;"Ñ"; FLASH 0;AT 20,11;"O->
"; FLASH 1;"º"; FLASH 0;AT 20,16
;"P->"; FLASH 1;"ª"; FLASH 0;AT
20,21;"C->"; FLASH 1;"¿"; FLASH
0;AT 20,26;"H->"; FLASH 1;"¡"; F
LASH 0
 100 LLIST : COPY : GO TO 9999
9000 REM Rutina caracteres españ
oles
9100 DIM a(11)
9105 DATA 65,69,73,79,85,78,80,8
1,67,72,77
9110 FOR i=1 TO 11: READ a(i): N
EXT i
9120 DATA 8,16,56,4,60,68,60,0
9125 DATA 8,16,56,68,120,64,60,0
9130 DATA 8,16,0,48,16,16,56,0
9135 DATA 8,16,56,68,68,68,56,0
9140 DATA 8,16,68,68,68,68,56,0
9145 DATA 24,0,60,34,34,34,34,0
9150 DATA 0,224,160,240,0,240,0,
0
9155 DATA 0,224,160,224,0,224,0,
0
9160 DATA 0,16,0,16,32,68,60,0
9162 DATA 0,16,0,16,16,16,16,0
9163 DATA 56,66,98,82,74,70,66,0
9165 FOR i=1 TO 11: FOR j=0 TO 7
9170 READ d
9175 POKE USR CHR$ a(i)+j,d
9180 NEXT j: NEXT i: RETURN
```

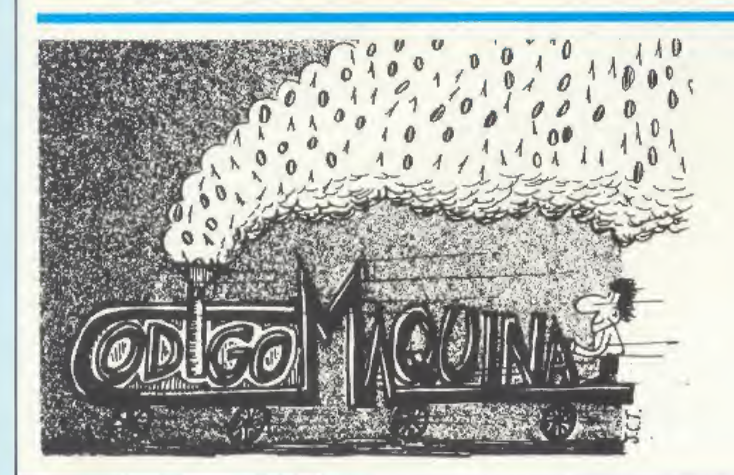
Cambio de base

34

Lo confesamos, aunque no nos guste presumir de ello, aunque reconocemos que para muchos carece de todo interés, aunque más de uno haya tenido graves problemas con ello... ¡nos gusta el código máquina! Y por ello sabemos mucho de los problemas con que se puede uno encontrar en la constante traducción de unas bases a otras. Este programa pretende facilitar un poco esta labor. Al igual que una calculadora, no es estrictamente necesario, pero puede ser muy útil.

Su manejo es muy sencillo. Al ejecutarlo aparece el menú de opciones en pantalla, pudiéndose trabajar en base binaria, decimal y hexadecimal.

Autor: José Félix 16 K




```
 10 REM
 20 REM ***********************
 30 REM JOSE FELIX ALVAREZ
        BIN-HEX 14/10/1984
 40 REM ***********************
 50 REM
 60 REM ***********************
 70 REM c$=Cabecera de bits.
 80 REM ***********************
 90 REM
100 LET c$="--------¦FEDCBA98765
43210¦--------"
```




```
 110 REM
 120 REM ***********************
 130 REM Introducion de datos.
 140 REM ***********************
 150 REM
 160 LET b$="": INPUT "Numero de
cimal ?  ";dec1: LET dec2=dec1
 170 REM
 180 REM ***********************
 190 REM Detecta el rango.
 200 REM ***********************
 210 REM
 220 IF dec1<0 OR dec1>65535 THE
N  PRINT #1; BRIGHT 1;"Error num
ero fuera de rango": BEEP 2,30:
PAUSE 50: GO TO 160
 230 IF dec1>=0 AND dec1<=255 TH
EN  LET bits=8: GO TO 300
 240 LET bits=16
 250 REM
 260 REM ***********************
 270 REM  Calcula >>> b$=Numero
         binario; h$=Numero hexa
         decimal.
 280 REM ***********************
 290 REM
 300 PRINT #1; BRIGHT 1; FLASH 1
;"Calculando resultados"
 310 FOR n=bits-1 TO 0 STEP -1
 320 LET digito=INT (dec1/2↑n)
 330 LET dec1=dec1-digito*INT (2
↑n)
 340 IF digito<1 THEN  LET b$=b$
+STR$ digito
 350 IF digito>=1 THEN  LET b$=b
$+CHR$ (digito+48)
 360 NEXT n
 370 IF bits=8 THEN  LET long=16
: GO TO 390
 380 LET long=8
 390 LET h$="": LET dec1=dec2
 400  FOR n=LEN (b$)/4-1 TO 0 ST
EP -1
 410 LET digito=INT (dec2/16↑n)
 420 LET dec2=dec2-digito*INT (1
6↑n)
 430 IF digito<10 THEN  LET h$=h
$+STR$ digito
 440 IF digito>=10 THEN  LET h$=
h$+CHR$ (digito+55)
 450 NEXT n
 460 REM
 470 REM ***********************
 480 REM Imprime resultados.
 490 REM ***********************
 500 REM
 510 PRINT c$
 520 PRINT  INK 7; PAPER 1; BRIG
HT 1;dec1;" d";: PRINT TAB long;
: FOR n=1 TO LEN b$
 530 IF b$(n)="0" THEN  PRINT  B
RIGHT 1; INVERSE 1;b$(n);: GO TO
 550
 540 PRINT  BRIGHT 1;b$(n);
 550 NEXT n
 560 PRINT "  "; PAPER 4; BRIGHT
 1;h$;" h": GO TO 160
```

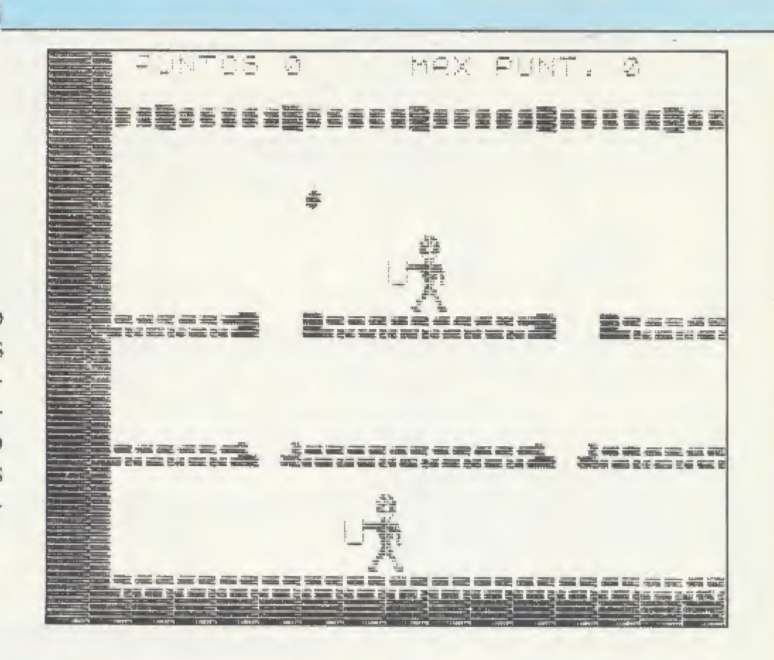
```
--------|FEDCBA9876543210|-------
10 d            00001010   0A h
--------|FEDCBA9876543210|-------
20 d            00010100   14 h
--------|FEDCBA9876543210|-------
50 d            00110010   32 h
--------|FEDCBA9876543210|-------
200 d           11001000   C8 h
--------|FEDCBA9876543210|-------
255 d           11111111   FF h
--------|FEDCBA9876543210|-------
1000 d  0000001111101000   03E8 h
--------|FEDCBA9876543210|-------
5000 d  0001001110001000   1388 h
--------|FEDCBA9876543210|-------
2500 d  0000100111000100   09C4 h
--------|FEDCBA9876543210|-------
25000 d 0110000110101000   61A8 h
```



PRIMERA REVISTA EN CASSETTE
SPECTRUM COMPUTING
¡Ya está a la venta! Cómprela en su quiosco habitual o solicítela a:
Bravo Murillo, 377-5° A
28020 MADRID
enviando el cupón que encontrará en esta revista
para 16K y 48K
875 ptas.
Como programar aventuras
Graficos tridimensionales
Jugando al Poker
El castillo embrujado
Misión suicida
64 columnas en su pantalla
y mucho mas
BIENVENIDO A
SPECTRUM COMPUTING
PRESENTA
BITMAN
COMO PROGRAMAR AVENTURAS
MISION SUICIDA
JUGANDO AL POKER
SCROLLER
GRAFICOS TRIDIMENSIONALES
EL CASTILLO ENDIABLADO
COWBOY
BANDIT
SKIING
64 COLUMNAS EN PANTALLA
GARANTIA
Calidad de carga asegurada

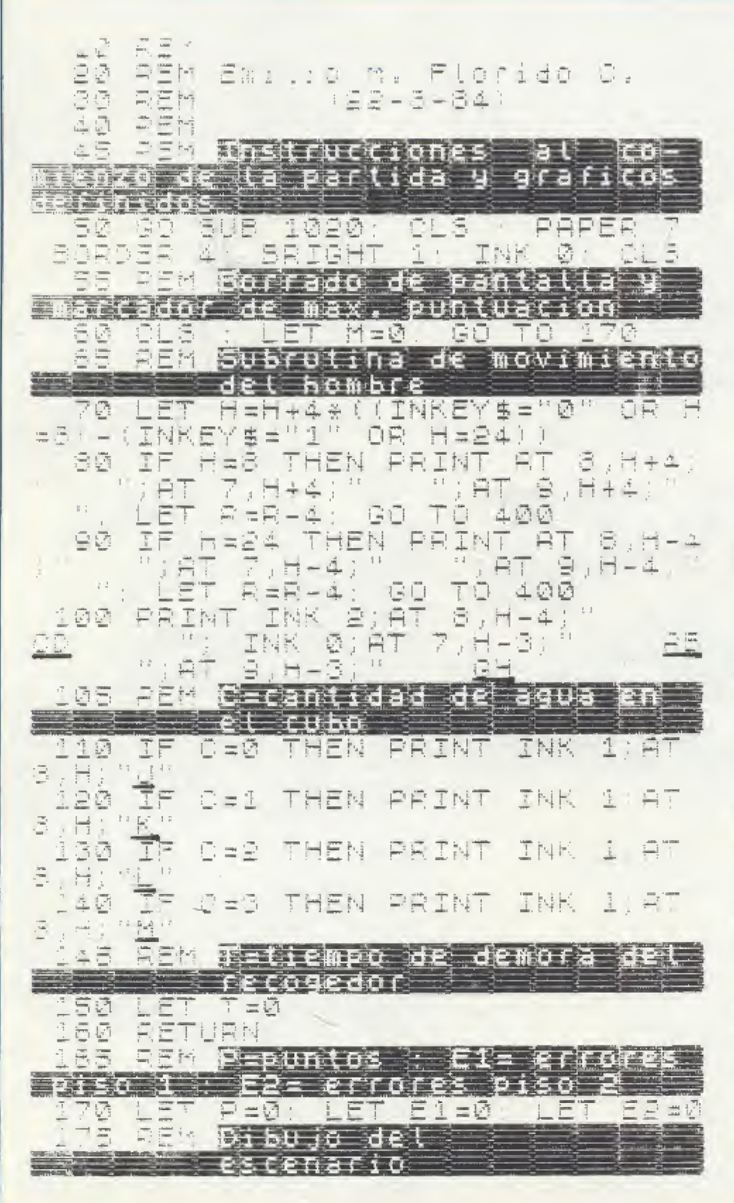
(35)

Goteras

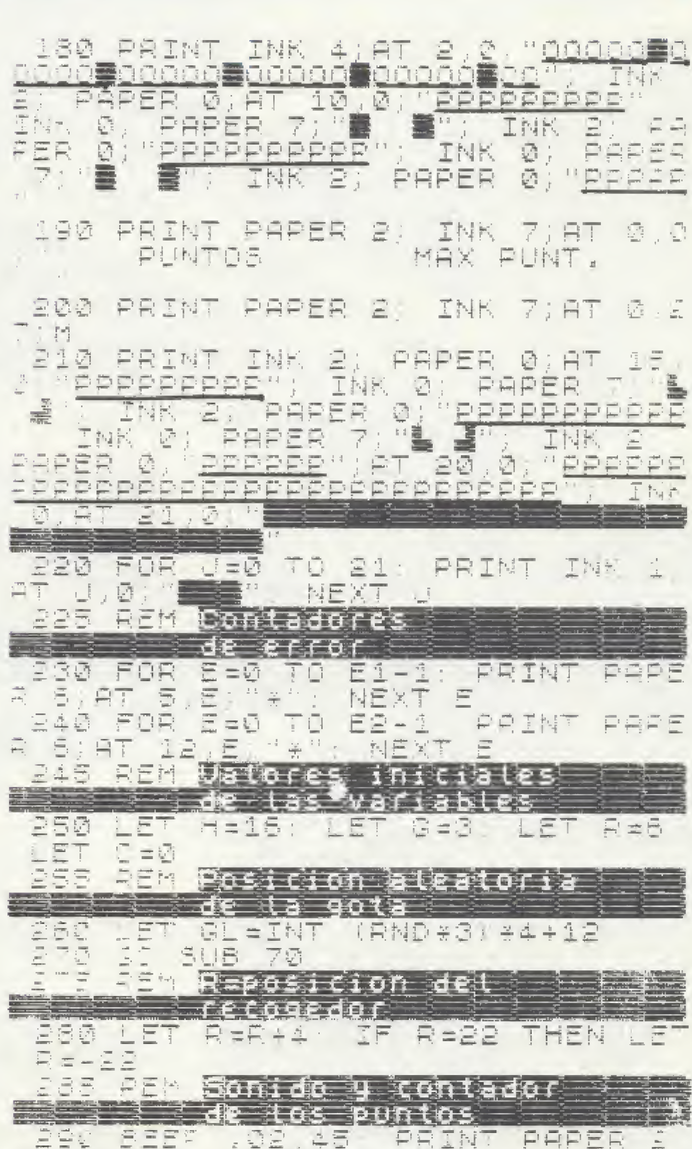


Ayudado de las teclas 0 y 1 podrá desplazarse de izquierda a derecha y a la inversa. El objetivo es evitar la inundación que a buen seguro ocurrirá si no se da prisa. Dado que las «gotas» tienen un tamaño considerable y que su «recipiente» no es demasiado grande, sólo caben tres gotas por viaje, que posteriormente habrá de pasar a su amigo del piso inferior. (Los caracteres gráficos aparecen subrayados).

Autor: Emilio Florido
16 K



```
  10 REM
  20 REM Emilio M. Florido C.
  30 REM       (22-3-84)
  40 REM
  45 REM Instrucciones al co-
mienzo de la partida y graficos
definidos
  50 GO SUB 1020: CLS : PAPER 7
 BORDER 4: BRIGHT 1: INK 0: CLS
  55 REM Borrado de pantalla y
marcador de max. puntuacion
  60 CLS : LET M=0: GO TO 170
  65 REM Subrutina de movimiento
          del hombre
  70 LET H=H+4*((INKEY$="0" OR H
=8)-(INKEY$="1" OR H=24))
  80 IF H=8 THEN PRINT AT 8,H+4;
"   ";AT 7,H+4;"   ";AT 9,H+4;"
  ": LET R=R-4: GO TO 400
  90 IF H=24 THEN PRINT AT 8,H-4
;"   ";AT 7,H-4;"   ";AT 9,H-4;"
  ": LET R=R-4: GO TO 400
 100 PRINT INK 2;AT 8,H-4;"
CD      "; INK 0;AT 7,H-3;"     EF
   ";AT 9,H-3;"     GH
 105 REM C=cantidad de agua en
          el cubo
 110 IF C=0 THEN PRINT INK 1;AT
8,H;"J"
 120 IF C=1 THEN PRINT INK 1;AT
8,H;"K"
 130 IF C=2 THEN PRINT INK 1;AT
8,H;"L"
 140 IF C=3 THEN PRINT INK 1;AT
8,H;"M"
 145 REM T=tiempo de demora del
          recogedor
 150 LET T=0
 160 RETURN
 165 REM P=puntos : E1= errores
piso 1 : E2= errores piso 2
 170 LET P=0: LET E1=0: LET E2=0
 175 REM Dibujo del
          escenario
```



```
 180 PRINT INK 4;AT 2,0;"00000█0
0000█00000█00000█00000█00"; INK
2; PAPER 0;AT 10,0;"PPPPPPPPP"
INK 0; PAPER 7;"█ █"; INK 2; PA
PER 0;"PPPPPPPPPP"; INK 0; PAPER
 7;"█  █"; INK 2; PAPER 0;"PPPP
"
 190 PRINT PAPER 2; INK 7;AT 0,0
;"   PUNTOS          MAX PUNT.
  "
 200 PRINT PAPER 2; INK 7;AT 0,2
7;M
 210 PRINT INK 2; PAPER 0;AT 15,
4;"PPPPPPPPP"; INK 0; PAPER 7;"▙
▟"; INK 2; PAPER 0;"PPPPPPPPPPPP
"; INK 0; PAPER 7;"▙ ▟"; INK 2;
PAPER 0;"PPPPPP";AT 20,0;"PPPPPP
PPPPPPPPPPPPPPPPPPPPPPPPPP"; INK
 0;AT 21,0;"████████████████████
█████████████"
 220 FOR J=0 TO 21: PRINT INK 1;
AT J,0;"███": NEXT J
 225 REM Contadores
          de error
 230 FOR E=0 TO E1-1: PRINT PAPE
R 5;AT 5,E;"*": NEXT E
 240 FOR E=0 TO E2-1: PRINT PAPE
R 5;AT 12,E;"*": NEXT E
 245 REM Valores iniciales
          de las variables
 250 LET H=16: LET G=3: LET R=6:
 LET C=0
 255 REM Posicion aleatoria
          de la gota
 260 LET GL=INT (RND*3)*4+12
 270 GO SUB 70
 275 REM R=posicion del
          recogedor
 280 LET R=R+4: IF R=22 THEN LET
 R=-22
 285 REM Sonido y contador
          de los puntos
 290 BEEP .02,45: PRINT PAPER 2
```



```
510 IF C=3 AND H=8 THEN PRINT I
NK 1;AT 12,10;"M"
515 REM Demora del recogedor
        segun la puntuacion
520 IF (R=-22 OR R=10) THEN LET
T=T+1
530 LET D=3: IF P>60 THEN LET D
=2: IF P>200 THEN LET D=1
540 LET R=R+4: IF T THEN LET R=
R-4: IF T=D THEN LET T=0: LET R=
R+4
550 IF R=22 THEN LET R=-22
560 IF R=-10 THEN LET R=10
565 REM Dibujo del
        recogedor
570 IF (ABS R=10 OR ABS R=14) T
HEN PRINT AT 18,-4+ABS R; INK 1;
"   J"; INK 4;"CD     ";AT 17,-4
+ABS R; INK 0;"     AB     ";AT 1
9,-4+ABS R;"     GH     "
580 IF (ABS R=18 OR ABS R=22) T
HEN PRINT AT 18,-4+ABS R; INK 1;
"   EF"; INK 1;"J     ";AT 17,-4
+ABS R; INK 0;"     AB     ";AT 1
9,-4+ABS R;"     GH     "
585 REM Eliminar fallos al
        llegar a 300 puntos
590 BEEP .02,45: FOR L=0 TO 5:
NEXT L: IF (P=300 OR P=301 OR P=
302) THEN LET E1=0: LET E2=0: PR
INT INK 1;AT 5,0;"   ";AT 12,0;"
   ": BEEP 1,40: BEEP .7,40: BEE
P 1,40: LET P=P+4
595 REM Rutina de vaciado del
        cubo a derecha o izquierda
600 IF INKEY$="1" AND H=8 THEN
GO TO 710
610 IF INKEY$="0" AND H=24 THEN
GO TO 790
620 IF INKEY$="0" AND H=8 THEN
PRINT AT 13,10;"  ";AT 14,11;"
";AT 12,10;"  ": GO TO 270
630 IF INKEY$="1" AND H=24 THEN
PRINT AT 13,22;"  ";AT 14,22;"
";AT 12,22;"  ": GO TO 270
640 GO TO 400
645 REM Se ha fallado en
        el piso 1
650 LET E1=E1+1
660 PRINT AT 5,12;"FALLASTE !":
IF E1=3 THEN PRINT PAPER 6;AT 5
,2;"*"
670 BEEP .5,10: BEEP .5,5: BEEP
1,1
680 FOR L=1 TO 150: NEXT L
685 REM Si cometemos 3 fallos
        fin del juego
690 IF E1=3 THEN GO TO 940
700 CLS : GO TO 180
705 REM Rutina de vaciado
        del cubo (izqd.)
710 PRINT AT 13,10;"N";AT 12,10
;" "
720 FOR J=1 TO C: FOR F=15 TO 1
7: BEEP .15,30
730 PRINT INK 1;AT F,10;"I";AT
F-1,10;" "
740 NEXT F: NEXT J
750 PRINT INK 1;AT 12,10;"J";AT
13,10;" "
755 REM El recogedor esta deba-
jo de donde vertemos el cubo ?
760 IF ABS R=10 THEN LET P=P+C:
LET C=0: GO TO 610
770 IF C=0 THEN GO TO 610
780 GO TO 870
785 REM Rutina de vaciado
        del cubo (derecha)
790 PRINT INK 1;AT 13,24;"N";AT
12,24;" "
800 FOR J=1 TO C: FOR F=15 TO 1
7: BEEP .15,30
810 PRINT INK 1;AT F,24;"I";AT
F-1,24;" "
820 NEXT F: NEXT J
830 PRINT INK 1;AT 12,24;"J";AT
13,24;" "
840 IF ABS R=22 THEN LET P=P+C:
LET C=0: GO TO 620
850 IF C=0 THEN GO TO 620
860 GO TO 870
865 REM Se ha fallado en
        el piso 2
870 LET E2=E2+1
880 PRINT AT 5,12;"CAYO FUERA !
"
890 BEEP .5,10: BEEP .5,5: BEEP
1,1
895 REM Si cometemos 3 fallos
        fin del juego
900 IF E2=3 THEN PRINT AT 12,2;
"*": GO TO 940
920 FOR L=1 TO 150: NEXT L
930 CLS : GO TO 180
935 REM Actualizar la max.
        punt. si procede
940 IF P>M THEN PRINT PAPER 2;
INK 7;AT 0,27;P: LET M=P
945 REM Rutina de fin de juego
        u otra partida
```


Programas/Juegos/Montajes/Còdigo Màquina
AÑO II - Núm. 15 - Febrero 1985 - 250 Ptas.
ZX
REVISTA PARA LOS USUARIOS DE ORDENADORES SINCLAIR
PROGRAMAS: BLACK-JACK y muchos más
YA ESTA A LA VENTA
BASIC para principiantes
Montaje: Reloj-digital para Spectr
SIMULADORES DE VUELO: en vivo y en el aire
¡GANE ESTE MONITOR!

```
950 FOR L=0 TO 150: NEXT L
960 PRINT AT 5,12;"SE ACABO !
"
970 BEEP 1,.1: BEEP 1,.5
980 PRINT AT 15,7;" OTRA PARTID
A? (S/N) "
990 IF INKEY$="s" THEN CLS : GO
 TO 170
1000 IF INKEY$="n" THEN BORDER 7
: BRIGHT 0: CLS : LIST 10: BORDE
R 4: BRIGHT 1
1010 GO TO 990
1015 REM Instrucciones al
 inicio de la partida
1020 CLS : PAPER 1: INK 7: BORDE
R 1: CLS
1030 PRINT AT 2,0;"  Se  encuent
ra  usted  en  una habitacion  p
or la que pasa una tuberia de la
 que caen goteras."
1040 PRINT AT 5,0;"  Dispone ust
ed de un cubo para recogerlas  y
  echarselas  a un amigo que est
a dos habitaciones mas abajo y q
ue se encargara de tirar el agua
."
1050 PRINT AT 10,0;"  Pero tenga
 cuidado  porque en su cubo  no
caben  mas de  tres goteras  y a
demas  su  amigo se esta moviend
o constantemente."
1060 PRINT AT 14,0;"  Utilice la
s  teclas "0" y "1" para  movers
e a  derecha  e iz- quierda  asi
 como para bajar de piso y solta
r el agua."
1070 PRINT AT 21,0;"Pulsa una te
cla para continuar"
1080 GO TO 1160
1090 FOR L=0 TO 21: PRINT INK 6;
AT L,0;"█ O O O O O O O O O O O
 O O █ "; BEEP .12,45
1100 BEEP .12,30: PRINT AT L,0;"

": NEXT L: CLS
1110 BEEP .5,25
1120 PRINT AT 2,0;"  Tenga  cuid
ado  de  no  fallarmas de  dos v
eces en la recogidade las gotera
s  o  al  verter elcubo pues sin
o perdera la parti-da."
1130 PRINT AT 7,0;"  La dificult
ad del  juego crecepaulatinament
e  y  si   consiguellegar a 300
 puntos como recom-pensa se le q
uitaran los erroresanteriores."
1140 PRINT AT 14,8;"BUENA SUERTE
 ! "
1150 PRINT AT 21,0;"  Pulsa una
tecla para empezar": GO TO 1190
1155 REM Graficos
 definidos
1160 FOR L=65 TO 80: FOR N=0 TO
7
1170 READ A: POKE USR CHR$ L+N,A
1180 NEXT N: NEXT L: PAUSE 0: GO
 TO 1090
1190 PAUSE 2: FOR L=0 TO 21: PRI
```



```
NT INK 6;AT L,0;"█ O O O O O O
 O O O O O O O O █"; BEEP .12,45
1200 BEEP .12,30: PRINT INK 6;AT
 L,0;"
      ": NEXT L: BEEP .5,25: CLS

1210 RETURN
1220 DATA 3,7,13,15,14,15,4,3:
REM A
1230 DATA 192,224,176,240,112,24
0,32,192: REM B
1240 DATA 3,255,255,15,7,7,3,3:
REM C
1250 DATA 192,248,252,252,236,23
6,204,204: REM D
1260 DATA 3,31,63,63,55,55,51,51
: REM E
1270 DATA 192,255,255,240,224,22
4,192,192: REM F
1280 DATA 3,7,7,14,28,24,120,120
: REM G
1290 DATA 192,224,224,112,56,24,
30,30: REM H
1300 DATA 8,8,28,62,126,126,124,
56: REM I
1310 DATA 129,129,129,129,129,12
9,129,126: REM J
1320 DATA 129,129,129,129,129,25
5,255,126: REM K
1330 DATA 129,129,129,255,255,25
5,255,126: REM L
1340 DATA 129,255,255,255,255,25
5,255,126: REM M
1350 DATA 126,129,129,129,129,12
9,129,129: REM N
1360 DATA 0,126,126,126,126,126,
126,0: REM O
1370 DATA 0,127,127,127,0,247,24
7,247: REM P
1380 STOP
1390 SAVE "GOTERAS" LINE 10
```

Preguntas y respuestas

P Les adjunto una nueva versión de la rutina de RESET publicada en el número 2 de su revista TODOSPECTRUM.

Esta versión sería compatible incluso para las máquinas de 16 K y se ubica en una sentencia REM al principio del programa.

Les envío mi modificación por si consideran conveniente publicarla en un futuro número de su revista, ya que lo considero de interés para muchos lectores.

Aitor García de Mardones
Vitoria

```
0000          00001 ;****  RESET   ****
0000          00002
0000          00003         ORG   23760
5CD0          00004
5CB2          00005 RAMTOP  EQU   5CB2H
5C3D          00006 ERRSP   EQU   5C3DH
5C71          00007 FLAGX   EQU   5C71H
5C3B          00008 FLAGS   EQU   5C3BH
0C0A          00009 POMSG   EQU   0C0AH
12A9          00010 MAIN1   EQU   12A9H
0D6E          00011 CLSLOW  EQU   0D6EH
1601          00012 CHOPEN  EQU   1601H
5CD0          00013
5CD0          00014
5CD0 3E19     00015 ACTIV   LD    A,25
5CD2 ED47     00016         LD    I,A
5CD4 ED5E     00017         IM    2
5CD6 C9       00018         RET
5CD7          00019
5CD7 52455345 00020 TEXTO   DEFM  'RESET exe
5CDB 54206578                     cuted'
5CDF 65637574
5CE3 6564
5CE5 A1       00021         DEFB  0A1H;'!' i
                                  nverso
5CE6          00022
5CE6 F5       00023 ENTRY   PUSH  AF
5CE7 3EFE     00024         LD    A,0FEH
5CE9 DBFE     00025         IN    A,(0FEH)
5CEB 1F       00026         RRA
5CEC 3807     00027         JR    C,NORES
5CEE 3EBF     00028         LD    A,0BFH
5CF0 DBFE     00029         IN    A,(0FEH)
5CF2 1F       00030         RRA
5CF3 3004     00031         JR    NC,RESET
5CF5 F1       00032 NORES   POP   AF
5CF6 FF       00033         RST   56
5CF7 ED4D     00034         RETI
5CF9          00035
5CF9 2AB25C   00036 RESET   LD    HL,(RAMTOP
                                  )
5CFC 2B       00037         DEC   HL
5CFD F9       00038         LD    SP,HL
5CFE 2B       00039         DEC   HL
5CFF 2B       00040         DEC   HL
5D00 223D5C   00041         LD    (ERRSP),HL
5D03 AF       00042         XOR   A
5D04 32715C   00043         LD    (FLAGX),A
5D07 CD0116   00044         CALL  CHOPEN
5D0A CD6E0D   00045         CALL  CLSLOW
5D0D 213B5C   00046         LD    HL,FLAGS
5D10 CB9E     00047         RES   3,(HL)
5D12 23       00048         INC   HL
5D13 CBEE     00049         SET   5,(HL)
5D15 AF       00050         XOR   A
5D16 11D65C   00051         LD    DE,TEXTO-1
5D19 CD0A0C   00052         CALL  POMSG
5D1C FB       00053         EI
5D1D C3A912   00054         JP    MAIN1
5D20 00       00055         NOP
5D21 00       00056         NOP
5D22 18C2     00057         JR    ENTRY
5D24          00058
5D24 3E3F     00059 DESACT  LD    A,63
5D26 ED47     00060         LD    I,A
5D28 ED56     00061         IM    1
5D2A C9       00062         RET
```

PROGRAMA CARGADOR

```
   1 REM xxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxx
  10 RESTORE : FOR i=23760 TO 23
851
  20 READ a: POKE i,a: NEXT i
  30 DATA 62,25,237,71,237,94,20
1,82,69,83,69,84,32,101,120,101,
99,117,116,101,100,161,245,62,25
4,219,254,31,56,7,62,191,219,254
,31,48,4,241,255,237,77,42,178,9
2,43,249,43,43,34,61,92,175,50,1
13,92,205,1,22,205
  40 DATA 110,13,33,59,92,203,15
8,35,203,238,175,17,214,92,205,1
0,12,251,195,169,18,0,0,195,230,
92,62,63,237,71,237,86,201
  50 POKE 23755,0: POKE 23756,0:
 STOP
```

R **El programa que nos envía está basado en el que apareció en el número 2 de TODOSPECTRUM sobre la creación de una tecla de RESET. Se puede usar en máquina de 16 K ya que el contenido del registro de interrupción I es menor que 40. Sin embargo, no podrá usarse con el interface I conectado, ya que almacena el programa al principio de la zona de BASIC.**

Para usar el programa, se tecleará el listado BASIC y después de ejecutarlo el REM se habrá convertido en línea 0, imposible de borrar o sobreescribir. Si borramos el resto del programa, podremos usarlo cuando deseemos haciendo uso de la instrucción MERGE.

Para ponerlo en funcionamiento escribiremos RANDOMIZE USR 23760 y para deasctivarlo, RANDOMIZE USR 23845.

P En la revista número 1, página 20 (Peek & pooke), variables 23552-23559 (Kstate) y refiriéndome al programa número 2 en la tecla SPACE el valor que me da a mí es de 32, no de 33 como publicáis vosotros. ¿El error es vuestro o es mío? Si es mío, ¿sabéis por qué y cómo lo puedo solucionar? Dispongo de un ordenador Spectrum 16 ampliado a 48 interiormente y el distribuidor es **Sitelsa.**

María Teresa Barcelona

R **En efecto, en el número 1 de TODOSPECTRUM se deslizó un error. Donde dice que el valor que corresponde a la tecla SPACE es el 33, debe decir 32, que es su valor en código ASCII. No tema, pues su Spectrum funciona perfectamente.**

Preguntas y respuestas

P Quería comentarles que el programa 64 caracteres del número 1 puede mejorarse bastante. Para ello he sustituido los datos de la dirección 32240 por 0,14,8,14,2,2, 14,0 y la 32496 por 0,0,14,8,14,2, 14,0, que son la «S» y la «s», respectivamente. Además he añadido:

```
   37 LET z=0: POKE 23658,Z
 1103 IF a=6 THEN LET Z=8*(Z=0):
POKE 23658,Z
```

con el fin de poder cambiar de mayúsculas y minúsculas con CAPS LOCK además de evitar confusiones al contestar a las preguntas. Por último, los ciclos de INPUT los he sustituido por sus homólogos con INKEYS, como se ve a continuación.

Santiago Cárdenas
Málaga

```
 110 PRINT #1;AT 1,0;"LEER O ESC
RIBIR?"
 120 IF INKEY$="l" THEN GO TO 40
0
 130 IF INKEY$="e" THEN GO TO 15
0
 140 GO TO 120
```

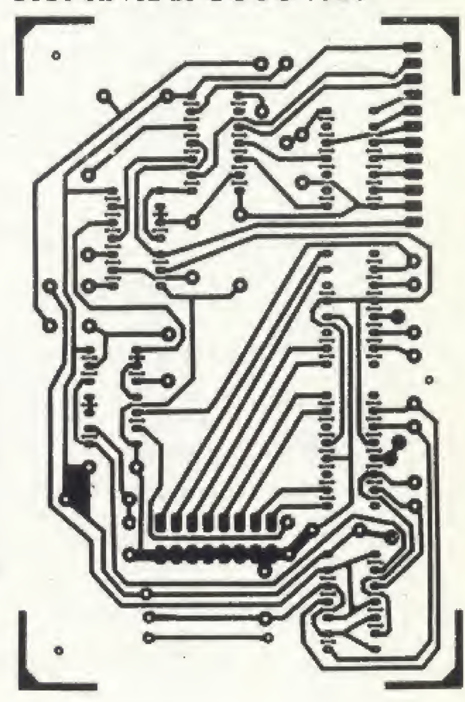
R **Procuramos hacer una rigurosa selección. No obstante, todo programa es susceptible de mejoras. Gracias.**

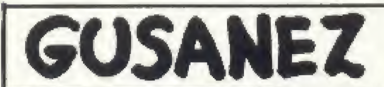
P Hemos realizado el circuito impreso del *interface* publicado en el número dos, encontrando un pequeño error en el circuito práctico, según el cual la pata 9 va unida a la 12 cuando debiera ser la 12 a la 8.

Daniel Molla
Toledo

R **Gracias a usted pudimos comprobar que se había deslizado un pequeño error en el circuito que se mostraba. Reproducimos el circuito impreso a que se hacía alusión en el artículo.**

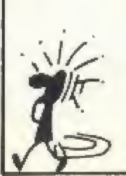
Así mismo, las instrucciones GOTO de los programas 2 y 3 de las líneas 1010 deben decir GOTO 1010 en vez de GOTO 7310.





por Jose C. Tomas







[P] Me he visto agradablemente sorprendido por la publicación de la entrevista con nuestros colaboradores programadores Joan Sales y Oscar Domingo. Quedaría muy agradecido que hicieseis constar que dicha entrevista tuvo lugar en nuestro local, que el equipo y material del que se habla en la entrevista y que aparece en las fotografías es de nuestra propiedad, y que está a disposición de todos los programadores con ideas que tengan ganas de colaborar con nosotros, y que pueden ponerse en contacto con nosotros personalmente, por teléfono o por escrito.

Josep-Oriol Tomás
Director de Ventamatic

[R] **Estamos seguros que más de uno le tomará la palabra y, por supuesto, el IBM-PC.**



[P] Me gusta el "montaje" del calendario de 1985, pero no me salen las cuentas. ¿Creía que diciembre siempre tenía 31 días?

Jose Antonio
Madrid

[R] **Ya hemos tomado medidas: ¡Hemos castigado a Gusánez a no tocar su nuevo Spectrum plus durante un mes! Pero antes le hicimos retocar la línea 130 para que después de diciembre colocase el número 31 en vez del 30.**

[P] Cuál es la dirección de la RUTINA (CALL) para limpiar toda la pantalla. En el ejemplar 2 de TODOSPECTRUM repiten la misma dirección para toda la pantalla que para el BORDER.

Antonio López
Manresa

[R] **En efecto en el artículo mencionado las rutinas de borrado de pantalla y de la parte inferior de esta tenían unas direcciones erroneas.**

Para eleminar los textos escritos en la parte inferior de la pantalla se llamará a la rutina de la dirección OD6EH (3438). La dirección de borrado de pantalla (CLS) es la OD6BH (3435). Unas rutinas muy cortas que pueden ser de gran interés debido a su velocidad son las siguientes:

```
0000               ;Borrado de imagen
0000
0000 01FF17        RUT1   LD     BC,17FFH
0003 21FF57               LD     HL,57FFH
0006 3600                 LD     (HL),0
0008 54                   LD     D,H
0009 5D                   LD     E,L
000A 1D                   DEC    E
000B EDB8                 LDDR
000D C9                   RET
000E
000E               ;Nuevos atributos
000E
000E 01FF02        RUT2   LD     BC,02FFH
0011 21FF5A               LD     HL,5AFFH
0014 3607                 LD     (HL),00000
                                 111B; ATTR
0016 54                   LD     D,H
0017 5D                   LD     E,L
0018 1D                   DEC    E
0019 EDB8                 LDDR
001B C9                   RET
```

El corcho

Una nueva «mini-sección» para que nuestros lectores puedan entablar contacto directamente entre sí. Los anuncios serán gratuitos, debiendo enviarse a TODOSPECTRUM, Bravo Murillo 377, 5.° A, indicando en el sobre «el corcho». No se admitirán anuncios de carácter publicitario.

VENDO ordenador Sinclair QL a estrenar provisto de 12 microdrives y cable para impresora. 110.000 pts. Teléfono (91) 446 88 29 horas de oficina. Lorenzo Bandrés.

OFERTA DE TRABAJO

Se solicita preparador de textos informáticos.
Su responsabilidad será la coordinación de la corrección, asesoramiento y preparación técnica de libros de microinformática.

CARACTERISTICAS

* *Titulado en Informática o Escuela Técnica*
* *Edad: 22 - 35 años*
* *Sólidos conocimientos de inglés*
* *Correcta redacción en castellano*

INTERESADOS

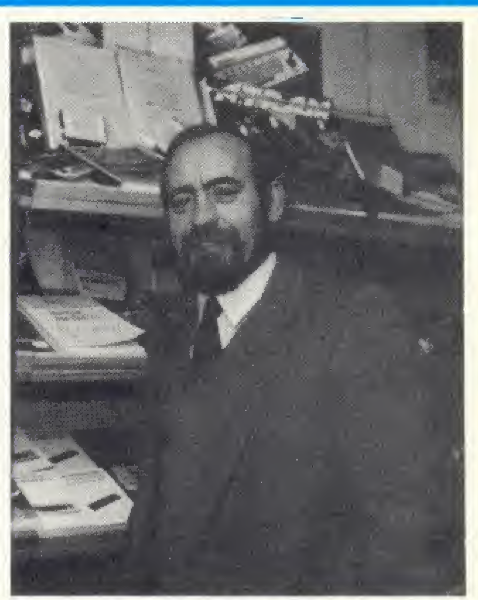
Llamar de 9 de la mañana a 5 de la tarde. (91) 245 82 05/04 (preguntar por la Sta. Olga)



ENTREVISTA

"El más vendido en los dos últimos meses es el Spectravideo, seguido del Spectrum y Commodore", comenta Antonio Sánchez de **Ada Computer-Zaragoza,** quien confiesa recomendar MSX a sus clientes "porque la informática tiene que unificarse y los japoneses van por este camino". Mariano Martínez, de **Sinclair Store-Madrid,** no se muestra tan convencido: "es absurdo gastarse 60.000 ptas., porque tenga una gama que le haga compatible con otra serie japonesa. Del MSX vendemos el Spectravideo porque es un ordenador muy serio de gestión y trabaja


Antonio Sánchez apuesta por el MSX.


Para Mariano Martínez el Spectrum sigue siendo la estrella.

Pasada la euforia de la compra de ordenadores domésticos efectuada en las navidades, dos expertos en ventas evalúan el mercado del Spectrum frente a la ofensiva de otros ordenadores.

con CPM, lo cual no ocurre con otros ordenadores MSX". Las ventas en esta tienda madrileña son favorables al Spectrum: "De cada diez ordenadores, siete son Spectrum (normal y plus) y el resto se reparte entre Amstrad, Spectravideo y Commodore, aunque este último está quedando absolutamente desplazado, en nuestra opinión porque no se ha cuidado la comercialización.

Es curioso detectar una cierta incertidumbre sobre el QL, cuya comercialización "oficial" a través de **Investrónica** se espera para estas fechas, una vez resueltos los problemas aparecidos en la primera versión. Los stocks de QL de los entrevistados parecen hacer prever buenos augurios para el nuevo ordenador de Sinclair, a pesar de la ofensiva nipona. Antonio Sánchez cree que son sectores distintos: "Los QL que hemos vendido han sido siempre para profesionales y sobre todo universitarios, en tanto el comprador del Spectravideo piensa generalmente en los juegos y aplicaciones para los chicos". En términos similares respondería su homólogo de Madrid: "El Spectravideo es un buen ordenador profesional, pero tiene un gran problema: su nombre. Lo primero que se piensa es que es un ordenador para jugar".

Por el momento el Spectrum sigue siendo el más vendido, aunque sólo sea, en muchos casos, simplemente por eso. Y para el más vendido, mayor número de programas para un público cada vez más maduro, como comenta Mariano Martínez: "Los padres demandan más programas didácticos y los chicos ya no se conforman con el simple juego de marcianos. El mercado ahora es más selectivo, pero entre unos y otros, la batalla la siguen ganando los adictos a los juegos. Sólo el 30 por 100 del *software* vendido para Spectrum, corresponde a aplicaciones".

Y para la primavera podemos prometer y prometemos mucho más. A la continuación del montaje sobre Lápiz óptico que permita su programación, y del Logo, para acabar de dominar la tortuga, destacamos:

— CODIGO MAQUINA. Nuestros lectores tienen la palabra.
— SINCLAIR QL: Comienza la ofensiva en España.
— SISTEMAS DE ALMACENAMIENTO DE DATOS EN SPECTRUM.



La versión española de Popular Computing

ORDENADOR POPULAR

LA REVISTA QUE INTERESA TANTO AL AFICIONADO COMO AL PROFESIONAL





Una publicación que informa con amenidad acerca de las novedades en el campo de las computadoras personales.

ORDENADOR POPULAR, la revista para el aficionado a la informática.

Ya está a la venta

Cómprela en su kiosco habitual o solicítela a:

ORDENADOR POPULAR

Bravo Murillo, 377
Tel. 7339662
28020 - *MADRID*

Programas

```
INK 7;AT 0,11;P
 300 IF R=-10 THEN LET R=10
 310 GO SUB 70
 315 REM Caida de
            la gotera
 320 LET G=G+1: IF G=10 THEN GO
TO 650
 325 REM Comprobacion de las go-
         teras que hay en el cubo
 330 IF C=4 THEN GO TO 650
 340 GO SUB 70
 345 REM Dibujo de
            la gotera
 350 PRINT INK 1;AT G,GL;"I";AT
G-1,GL;" "
 355 REM Dibujo del
            recogedor
 360 IF (ABS R=10 OR ABS R=14) T
HEN PRINT AT 18,-4+ABS R; INK 1;
" I"; INK 4;"CD    ";AT 17,-4
+ABS R; INK 0;"    AB    ";AT 1
9,-4+ABS R;"    GH    "
 370 IF (ABS R=18 OR ABS R=22) T
HEN PRINT AT 18,-4+ABS R; INK 4;
"    EF"; INK 1;"J    ";AT 17,-4
+ABS R; INK 0;"    AB    ";AT 1
9,-4+ABS R;"    GH    "
 375 REM Comprobar si se
            recoge la gotera
 380 IF G=8 AND GL=H THEN LET G=
3: BEEP .12,C*10+20: LET C=C+1:
LET P=P+1: LET G=G+1: GO TO 260
 390 GO TO 270
 395 REM Si recogedor pasa por
el centro bajar la gota un lugar
 400 IF ABS R=18 THEN LET G=G+1:
IF G=10 THEN GO TO 650
 405 REM Caida de
            la gotera
 410 PRINT AT G-1,GL;" "; INK 1;
AT G,GL;"I";AT G-1,GL;" "
 415 REM Bajar el hombre
            al piso 2
 420 IF H=8 THEN PRINT INK 2;AT
13,11;"CD"; INK 0;AT 14,11;"GH";
AT 12,11;"AB"
 430 IF H=24 THEN PRINT INK 2;AT
13,22;"EF"; INK 0;AT 14,22;"GH"
;AT 12,22;"AB"
 432 REM Comprobacion de las go-
         teras que hay en el cubo
 435 IF C=4 THEN GO TO 650
 438 REM Dibujar el cubo lleno
       hasta donde corresponda
 440 IF C=3 AND H=24 THEN PRINT
INK 1;AT 12,24;"M"
 450 IF C=0 AND H=8 THEN PRINT I
NK 1;AT 12,10;"J"
 460 IF C=0 AND H=24 THEN PRINT
INK 1;AT 12,24;"J"
 470 IF C=1 AND H=8 THEN PRINT I
NK 1;AT 12,10;"K"
 480 IF C=1 AND H=24 THEN PRINT
INK 1;AT 12,24;"K"
 490 IF C=2 AND H=8 THEN PRINT I
NK 1;AT 12,10;"L"
 500 IF C=2 AND H=24 THEN PRINT
INK 1;AT 12,24;"L"
```